AN FPGA BASED DIGITAL MODULATION CLASSIFIER

BY

MUHAMMAD SAAD SADIQ

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

JUNE 2011

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Muhammad Saad Sadiq** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING.**

Thesis Committee

Dr. Ali A. Al-Shaikhi
Department Chairman

Dr. Mohamed Deriche
(Advisor)

Dr. Salam A. Zummo
Dean of Graduate Studies

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dr. Alaa El-Din Hussein
(Member)

19th June, 2011
Date

Dr. Abdul-Hafid Bouharaoua
(Member)

AN FPGA BASED DIGITAL MODULATION CLASSIFIER

by

MUHAMMAD SAAD SADIQ

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

In Partial Fulfillment of the Requirements
for the degree

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

KING FAHD UNIVERSITY
OF PETROLEUM & MINERALS

Dhahran, Saudi Arabia

JUNE 2011

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **MUHAMMAD SAAD SADIQ** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**.

Thesis Committee

Dr. Mohamed Derich (Adviser)

Dr. Alaa El-Din Hussein (Member)

Dr. Abdul Hafid Bouharaoua (Member)

Dr. Ali A. Al-Shaikhi
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

Date

Dedicated to my loving Family

ACKNOWLEDGMENTS

In the name of Allah, the Most Beneficent, the Most Merciful

All praise be to Allah (The One and The Only Creator of everything) for His limitless blessings. May Allah bestow peace and His choicest blessings on His last prophet, Hazrat Muhammad (Peace Be Upon Him), his family (May Allah be pleased with them), his companions (May Allah be pleased with them) and his followers.

I would like to express my profound gratitude and appreciation to my thesis committee chairman and adviser, Dr. Mohamed Deriche, whose expertise, understanding, and patience, added considerably to my graduate experience. I appreciate his vast knowledge and skill in many areas e.g., digital signal processing, modeling, pattern recognition, communication systems. Dr. Deriche is the one professor who truly made a difference in my life. It was under his tutelage that I developed a focus and became interested in research. He provided me with direction, intellectual support and became more of a mentor than a professor. It was through his persistence, understanding and kindness that I completed my degree. I doubt that I will ever be able to convey my appreciation fully, but I owe him my eternal gratitude.

I am grateful to Computer Engineering department's Professor Dr. Abdul Hafid Bouharaoua for providing me the opportunity to work with him on this thesis project. Without his support and understanding, this thesis work would have been impossible. I would also like to thank my comittee member Dr. Alaa El-Din Hussein for the assistance he provided at all levels of the thesis research.

Finally, I would like to thank the entire committee for their useful suggestions and for taking out time from their busy schedules to serve as committee members.

I would also like to thank my friends, particularly Faraz Ahmed, Babar Hasan, Umer bin Saeed, Khaja Wahab, Jamal Khan, Muhammad Asad, Zeeshan Rizvi, Sarfraz Abbasi, Saqib Sohail, Zeehasham, Salman Ahmed, Ahmed AQ and Syed Wajahat for our exchange of knowledge, ideas, skills, and extra cirricular activities during Master's studies, which helped enrich the experience.

I would also like to thank my family for the support they provided me throughout my entire life and in particular, I must acknowledge my father, Dr. Muhammad Sadiq, and my mother. Without their love and encouragement, I would not have accomplished anything worthwhile in my career.

In addition, I acknowledge that this reserach would not have been possible without the support and assistance provided by the King Fahd University of Petroleum and Minerals.

TABLE OF CONTENTS

LIST OF TABLES viii

LIST OF FIGURES x

ABSTRACT (ENGLISH) xiii

ABSTRACT (ARABIC) xv

NOMENCLATURE xvi

1 INTRODUCTION 1

1.1 Introduction to Communication systems 2

1.2 Modulation of Communication Signals 3

1.2.1 Amplitude Modulation (AM) 4

1.2.2 Frequency and Phase Modulation (FM & PM) 6

1.3 Digital Communication 7

1.3.1 Aplitude Shift Keying (ASK) 10

1.3.2 Phase Shift Keying (PSK) 12

1.3.3 Frequency Shift Keying (FSK) 13

1.4 Communication in Military / Software Radio 15

1.5 Pattern Recognition of Communication Signals 16

1.5.1 Types of modulation classifiers: 19

1.6 Classification of Digital Modulation Schemes 22

1.7 Blind / semi blind classification techniques 24

1.8 Problem Statement 25
1.9 Objectives of the thesis 26

2 LITERATURE REVIEW 28
2.1 Main Approaches used for Modulation Classification 28
2.1.1 Likelihood-based (LB) approach 29
2.1.2 Feature based approach 36
2.2 Modulation Classifiers Implemented on Hardware 43
2.2.1 Digital Signal Processing (DSP) kits 45
2.2.2 Field Programmable Gate Array (FPGA) 47
2.2.3 General Purpose Processors (GPP) 48
2.2.4 Rapid Radio devices 49

3 THE PROPOSED MODULATION CLASSIFICATION FRAMEWORK 51
3.1 Basic digital modulation classifier framework 52
3.2 Selecting Features 53
3.2.1 Statistical Moments 56
3.3 Selecting Classifier 58
3.3.1 The Bayesian Classifier 60
3.4 The complete proposed model 62
3.4.1 Signal Generation 63
3.4.2 System Training 63
3.4.3 Testing Unknown Data 68
3.5 Hardware Implementation framework 69
3.5.1 Selecting suitable platform 70
3.5.2 Objectives of the Implementation 73
3.5.3 Integral components of an FPGA 73
3.5.4 Area, power and speed tradeoffs 76
3.5.5 Parallel Processing 77
3.5.6 Necessity of Floating point operations 77

4 TESTING AND EVALUATION OF THE PROPOSED SYSTEM 82

4.1 Classification accuracy 83

4.2 Handling uncertainties in classification by thresholding 85

4.3 Gaussianity Test 88

4.3.1 Jarque bera test 89

4.3.2 Lillie fors test 91

4.4 Robustness of the Selected Features 91

4.4.1 Neural Networks Classifier 92

4.4.2 Nearest Neighbor Algorithm (NN) 93

4.5 Different Effects degrading the System Performance 94

4.5.1 Channel Modeling 94

4.5.2 Fading effects 96

5 TESTING AND EVALUATION OF THE PROPOSED SYSTEM 103

5.1 Designing the Architecture 104

5.1.1 FSM based architecture 104

5.1.2 Multi-core Architecture 108

5.2 Components of the Architecture 111

5.2.1 Dual Port Instruction Ram 112

5.2.2 Dual port & Dual module Data RAM 112

5.2.3 Master Control Register 114

5.2.4 Floating point Unit (FPU) 116

5.2.5 Mixed Signal Implementation 117

5.3 Instruction set 119

5.3.1 Increment 119

5.3.2 Loop 120

5.3.3 Zero Crossing Count 120

5.3.4 Probability comparison 123

5.4 Target Device Parameters 125
5.5 Algorithm constraints 127
5.5.1 Memory Requirement 127
5.5.2 Recursive Equations 128
5.6 Synthesis & Implementation 128
5.6.1 Power Utilization 133
5.6.2 Verification of the hardware 134
5.6.3 Design Summary 134

6 DEVELOPMENT OF THE GUI 139
6.1 Serial Interfacing 139
6.1.1 UART Receiver (Rx) 141
6.1.2 UART Rx Finite State Machine (FSM) 141
6.1.3 UART Transmitter (Tx) 143
6.1.4 First In First Out (FIFO) buffer 145
6.1.5 Hyper-Terminal 145
6.2 Selecting platform for software development 147
6.2.1 Java Development Kit (JDK) 147
6.2.2 C++ 148
6.2.3 Matlab 149
6.3 Developed Graphical User Interface (GUI) 149

7 CONCLUSIONS 152
7.1 Summary of Contribution 153
7.2 Future Direction 154
7.3 Conclusion 155

REFERENCES 158

VITAE 173

LIST OF TABLES

3.1 Common platforms for the implementation of modulation classifiers 72

4.1 Contingent Classification without Thresholding at SNR 10dB . . . 90
4.2 Contingent table at 10dB SNR after applying zero crossing and AIC thresholds . . . 91
4.3 Performance (%) of the classifier using ZC and AIC thresholds . . 91
4.4 Contingent table - results of ANN classification with SNR=5dB . 92
4.5 Contingent table - results of k-NN classification with SNR=5dB . 93
4.6 Terrain Type and Doppler Spread for SUI Channel Models 99
4.7 SUI - 2 Channel Model . . . 99
4.8 SUI - 3 Channel Model . . . 100
4.9 Contingent table of classification under 1-tap Rayleigh fading channel 101
4.10 Contingent table of classification under 2-tap Rayleigh fading channel 101
4.11 Contingent table under Rayleigh fading channel and thresholds . . 102

5.1 Clock cycle requirement for available arithmetic operations 116
5.2 Performance Comparison to competing FPU by Usselmann . . . 116
5.3 Conditions for Zero crossing count . . . 123
5.4 Summary of the Spartan-3E FPGA attributes . . . 126
5.5 Device Utilization of the proposed FPGA architecture . . . 129
5.6 Timing Report for Spartan 3E-1200e FG320 . . . 131
5.7 Power required for the target . . . 134
5.8 Percentage of identification of the proposed modulation classifier at 10dB SNR . . . 137

5.9 Summary of the Spartan-3E FPGA attributes 137
5.10 Summary of the Spartan-3E FPGA attributes 137

LIST OF FIGURES

1.1 Basic building blocks of the communication system 5
1.2 Signal waveform of Amplitude Modulation 5
1.3 Signal waveform of frequency modulation 9
1.4 Separation of transmitting process into two coding blocks 9
1.5 Basic block diagram of the digital communication system 11
1.6 Signal waveform of the Amplitude Shift Keying modulation 11
1.7 Signal waveform of the Phase Shift Keying modulation 14
1.8 Signal waveform of the Frequency Shift Keying modulation 14
1.9 Placement of the modulation recognizer in the receiver architecture 18
1.10 Different terminologies related to pattern recognition 18
1.11 Block diagram of a generalized classifier 20
1.12 frequency vs standard deviation plot for gender classification using speech 20
1.13 Placement of the modulation classification at the receiver 23
1.14 Different divisions of modulation classification 23

2.1 Basic approaches used in modulation classification 31
2.2 Block diagram of the Maximum Likelihood classification 31

3.1 A general configuration of a modulation classifier 54
3.2 Commonly used classification methods and their respective feature sets 54
3.3 Overview of the proposed system framework 64
3.4 Generated training signal waveform 64

3.5 Extracting features from the training signal 67
3.6 OVA arrangement for the test case of ASK4 classification 67
3.7 OVA arrangement for all 6 modulation schemes 67
3.8 Unknown signal waveform 75
3.9 Layout of Configurable Logic Blocks inside an FPGA 75
3.10 Example of parallel processing using multiple modules at same time 78
3.11 Frame of an IEEE754 floating point unit data value 81

4.1 ROC of the proposed modulation classifier 88
4.2 Features extracted from ASK modulation 90
4.3 features extracted from FSK modulation 90
4.4 features extracted from PSK modulation 90
4.5 Multipath propagation in an urban environment 102

5.1 Block diagram of the implementation phase 105
5.2 Design flow of FPGA implementation 105
5.3 Abstract illustration of FSM based architectures 107
5.4 Data path of the designed FSM based architecture 107
5.5 Proposed Multi-execution architecture 110
5.6 Structure of the Instruction RAM elements 113
5.7 Block diagram of the true dual port instruction memory 113
5.8 Design of the Duplicated true dual port data ram 115
5.9 Structure of the Data Ram elements 115
5.10 Structure of the Master control register 115
5.11 The FPU architecture by Al-Eryani 118
5.12 Different instructions developed for the proposed architecture . . . 121
5.13 Hardware designed for the increment instruction 121
5.14 Block diagram of describing the Loop instruction 122
5.15 Loop reset condition and direct manipulation of the program counter 122
5.16 Sample instruction describing the zero crossing operation 124
5.17 Block diagram of the final probability comparison 124

5.18 Block diagram of the Nexys2 development kit 129
5.19 Top level block level schematic of the proposed architecture 131
5.20 Device level schematic for the generated model 132
5.21 The floor planning for the proposed model 135
5.22 Power vs Voltage curve of the proposed architecture 135
5.23 Power vs Temperature curve of the proposed architecture 136
5.24 Power variance of the proposed architecture 136
5.25 The power division with respect to function 138

6.1 Frame of the serially transmitted data 142
6.2 UART receiver system 142
6.3 FSM state and functionality diagram of the receiver 144
6.4 Complete UART system 146
6.5 Structure of the Data RAM elements 146
6.6 Structure of the Instruction RAM elements 146
6.7 Designed GUI for the PC to FPGA interface 151

THESIS ABSTRACT

NAME: Muhammad Saad Sadiq

TITLE OF STUDY: An FPGA based Digital Modulation Classifier

MAJOR FIELD: Electrical Engineering

DATE OF DEGREE: June 2011

The problem of classifying the modulation scheme of an unknown signal has captured much attention in the field of signal processing and communications. It is due to the nature of this problem that finds application in many defense areas like, electronic warfare, military threat analysis, or electronic counter-counter measure. Also in many civilian based technologies like signal confirmation, spectrum management, interference identification monitoring. The primary direction of the thesis is towards the classification of signals with unknown modulations and no a-priori knowledge using simple Bayes classifier. In order to classify the signals, four statistical moments were selected namely the mean, variance, skewness and kurtosis. Six modulation schemes were classified namely, 2,4-ASK, 2,4-PSK and 2,4-FSK. Modulation classification was performed under degrading channel con-

ditions with additive white Gaussian noise (AWGN) and Rayleigh fading channel effects. Results indicate that the proposed classifier has high identification accuracy at low SNR and with fading effects.

Second part of the thesis implements the automatic modulation identification framework on Xilinx Spartan XC1200Efg320 FPGA device. A generic architecture is developed to serve as a testbed for verifying DSP algorithm on hardware. The developed architecture on FPGA can be serially scheduled and the operations can be performed in parallel. The architecture is pipelined and can be scaled up or down according to the application or the resources. The synthesis and test results show that the proper operation of the algorithm was replicated on the FPGA device. A low power, low area and highly accurate modulation classifier was verified with real time performance.

Nomenclature

Abbreviations

ASK	:	Amplitude Shift Keying
PSK	:	Phase Shift Keying
FSK	:	Frequency Shift Keying
AWGN	:	Additive White Gaussian Noise
FPGA	:	Field Programmable Gate Array
AM	:	Amplitude Modulation
FM	:	Frequency Modulation
PM	:	Phase Modulation
MSK	:	Minimum Shift Keying
GSM	:	Global System for Mobile Communication
SDR	:	Software Defined Radio
CR	:	Cognitive Radio
BER	:	Bit Error Rate
AMI	:	Automatic Modulation Identification
OFDM	:	Orthogonal Frequency Division Multiplexing
SNR	:	Signal to Noise Ratio
PDF	:	Probability Density Function

Notations

A_c	:	Amplitude of the carrier signal
ω_c	:	Carrier Frequency
Φ_Δ	:	Instantaneous Phase
$s[m]$	:	discrete symbol sequence
$u(t)$	:	unit step function
Φ_X	:	Characteristic Function
Ψ_Z	:	Cumulative Function
$r(t)$	:	Received Signal
m_n^X	:	nth order moment of X
u	:	Mean of the Sample distribution
σ^2	:	Variance of the Sample Distribution
γ	:	Skewness of the sample distribution
κ	:	Kurtosis of the sample distribution
$\|R\|$	:	determinant of an l x l covariance matrix
$R_{xx}(k)$	:	Autocorrelation estimate
N	:	Total number of samples in the distribution

CHAPTER 1

INTRODUCTION

This section briefly describes the communication system and the relative modules used in our work. A general overview of the communication system, different types of modulations and their definitions are given. The modulation schemes used in the proposed work are also defined. Later in this section, a motivation is developed that outlines the problem of modulation classification. Also, classification methodologies and their characteristics are described. After the motivation and a brief background are developed, the thesis problem is stated. A proposed solution is then suggested and thesis objectives are derived from it.

Later sections of the thesis are organized into 6 chapters and one appendix, excluding the introduction chapter. Chapter 2 reviews a comprehensive literature survey on the problem of modulation classification. Chapter 3 and 4 present the proposed modulation classifier, its performance summary and overall framework. Chapter 5 contains the practical implementation and hardware verification of the proposed modulation classifier. Chapter 6 details the graphical user interface de-

velopment that assists in the verification process. Finally, chapter 7 contains the conclusion, future research venues related to the proposed work and the references.

1.1 Introduction to Communication systems

A typical communication system starts with the concept of its source of information, which outputs the information of interest that is to be sent somewhere. This source of information is given to the system, called the transmitter. The job of a transmitter is to prepare the information for actual sending through some physical medium.

In a sense the transmitter matches the characteristics of the signal to the characteristics of the medium. The physical medium through which the data is transmitted is called the channel. The channel is an abstraction for every kind of physical medium through which information is transmitted [1].

At the receiver end, the job is to ensure that the information is given to the user in an appropriate form that can be used in some application. A communication system has the following five major blocks, as shown in fig. 1.1.

There can be many kinds of information sources, the most common form of source is voice and data signals. These information sources have their own set of requirement for the communication system.

One of the features of the information sources is the spectral band or spectral occupancy that is directly proportional to the bandwidth of the channel. Bandwidth is a very important attribute or resource of the communication system.

Sending raw analog voice signals in its original form is quite impossible because the antenna size would be very large [2]. Also, when dealing with a large number of signals, it's very difficult to allocate bandwidth to all of them. Some frequency bands may hive higher signal attenuation then others, thus a user should be able to switch to higher frequency bands. Hence to overcome the above stated problems; the transmitter has to perform a specific process called modulation.

1.2 Modulation of Communication Signals

In a communication system, the transmitted signal has to propagate through a channel. A communication channel is a physical environment having a finite band-width. The communication channel is also divided into further sub-divisions to allow multiple users to transmit [1]. This limited bandwidth of the channel restricts the band-width of the transmitted signal. We have to first embed voice signal into some other signal with higher frequency so the antenna size requirement would become much smaller. Hence we reach at the idea of a carrier signal. The job of the carrier is to carry the information signal along with it. This process of combining the carrier wave and the information signal is called modulation. The amplitude, phase, or frequency of this carrier is altered proportionally to the transmitted information signal. This operation is known as modulation.

Modulation of a sine wave is done to convert a low frequency message signal into a high frequency transmission signal, for example an audio signal could be converted into a radio-frequency signal (RF signal).

The types of modulation can be sub-divided into two categories depending on the type of final transmitted signal. If the transmitted signal is continuous, it is called analog modulation. If the transmitted signal consists of a finite alphabet of discrete symbols, it is called digital modulation.

1.2.1 Amplitude Modulation (AM)

Amplitude modulation is the simplest modulation scheme. AM is achieved by changing the amplitude of the carrier wave with respect to the modulating signal, as shown in figure 1.2.

The analytic representation of the modulated signal can be expressed as

$$z(t) = A_c[1 + \mu x(t)]e^{jw_c t} \tag{1.1}$$

i.e. a sum of the carrier signal and the modulating signal scaled by $A_c u$ and shifted in frequency by w_c. The resultant bandwidth 'B' of the modulated signal is $2w_c$ [3]. The envelope of the AM signal can be express as

$$a(t) = A_c[1 + \mu x(t)] \tag{1.2}$$

i.e. the modulating signal is biased to positive values. The instantaneous phase can be expressed as

$$\phi(t) = \tan^{-1}\left[\frac{A_c[1 + \mu x(t)]\sin(w_c t)}{A_c[1 + \mu x(t)]\cos(w_c t)}\right] = w_c t \tag{1.3}$$

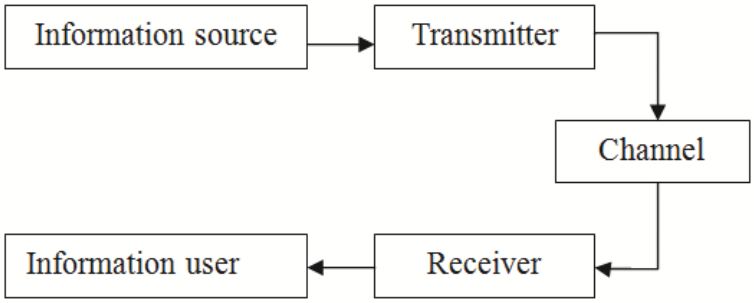

Figure 1.1: Basic building blocks of the communication system

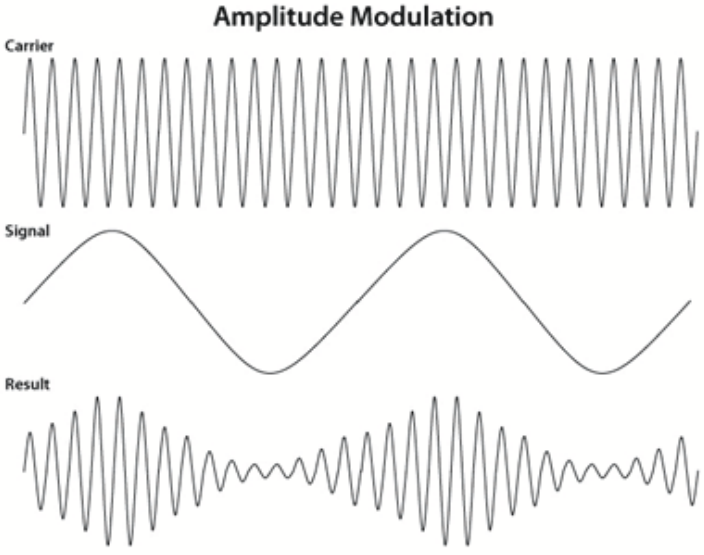

Figure 1.2: Signal waveform of Amplitude Modulation

The instantaneous frequency is simply

$$f(t) = \frac{1}{2\pi}\frac{d(w_c t)}{dt} = f_c \tag{1.4}$$

1.2.2 Frequency and Phase Modulation (FM & PM)

In the FM and PM modulation schemes the instantaneous amplitude remains constant and the angle of the carrier is altered. In FM the instantaneous frequency is altered according to the modulating signal [3], as shown in figure 1.3. The analytics FM signal may be expressed as

$$z(t) = A_c exp\left\{j[w_c t + 2\pi f_\Delta \int_{t_0}^{t} x(\lambda)d\lambda]\right\} \tag{1.5}$$

In phase modulation the instantaneous phase is varied according to the modulating signal. The analytic phase modulated signal may be expressed as

$$z(t) = A_c exp\left\{j[w_c t + \phi_\Delta x(t)]\right\} \tag{1.6}$$

In a communication system, the phenomenon of noise limits the transmission of data through a channel. In the absence of noise we can derive infinite precision and thus no constraint on the amount of data that we can send [4]. However, noise is uniformly present throughout different time and frequency measures.
The most common type of noise model used is the Additive white Gaussian noise AWGN. Noise can be due to many internal and external system sources. When

we combine the effect of all noise sources, then according to central limit theorem, this combined noise will be Gaussian and can be estimated using additive white Gaussian noise. White noise means that the noise is uncorrelated at any other point in time.

For the additive white Gaussian noise with a bandwidth constraint, the capacity to send data becomes

$$C = Wlog_2\left(1 + \frac{P}{N_0 W}\right) \tag{1.7}$$

Where W is the bandwidth, P is the signal power. This is the ultimate, and the maximum achievable bit rate in practice [1]. This equality presents the fastest that a communication system can transmit.

1.3 Digital Communication

The reason to communicate in digital medium is that digital systems are much easier to design and implement, increasing the practicality of the system. Digital systems are also very cheap and can be produced on mass scale, reducing the production cost.

In digital communication the data to be transmitted is in digital format. If the information is started to be analog, it is first converted to digital by analog to digital converters. This is done through the process of sampling and quantization. In the case of digital data, the data is aligned as a sequence of bits that can be transmitted with little manipulation. A digital signal is characterized by the fact

that every point and every sample can take only finite number of values, unlike in analog signals [4].

In the process of communicating data across a channel, Shannon proposed a model that dissects the transmitter block into two separate processes, as shown in figure 1.4. The primary transmitting process will compress the information source to its maximum and the second block will send the compressed data into an efficient manner. The first block is called source coding, while the latter is known as channel coding. In short, Source coding relates to the source while the channel coding block relates to the channel. By dividing the transmitter block, we don't lose any efficiency or effectiveness.

The source encoder compresses the information binary source into minimum number of bits to save bandwidth. Bandwidth is an important commodity and all redundant bits are suppressed to minimize the data that will use the bandwidth. The channel encoder shapes the data bits into an appropriate way that all the bits are propagated without any information loss. In a way its job is to match the characteristics of the data bits to that of the channel. The major advantage of this block is that if any error is introduced by the channel in the data stream, then the channel decoder will be able to correct that error to some extent.

Many channels are analog in nature and hence we need the digital modulator that converts the digital data into analog form in order for it to be transmitted on the analog channel.

The receiver parts work in the opposite manner to retrieve the information from

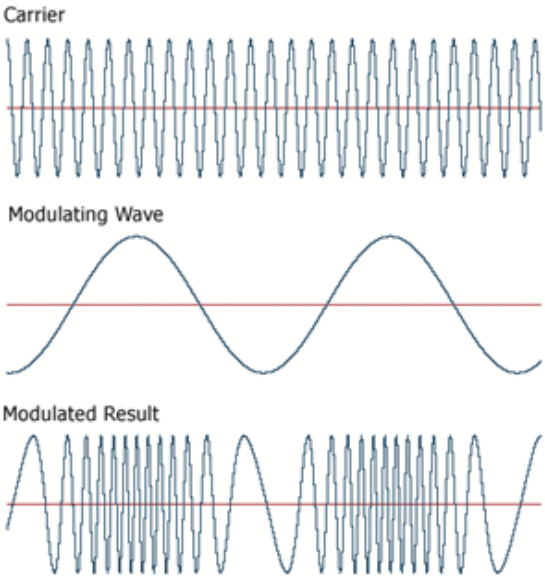

Figure 1.3: Signal waveform of frequency modulation

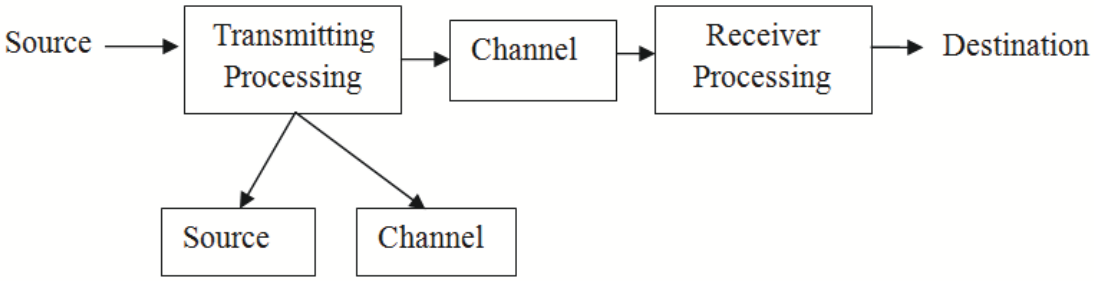

Figure 1.4: Separation of transmitting process into two coding blocks

the transformed bits and results in the output bits. A general analytic representation of digital modulated signals is given by

$$z(t) = A_c e^{jw_c t} \sum_{m=-\infty}^{\infty} s[m]g(t - mT) \tag{1.8}$$

Where A_c is the amplitude and w_c is the frequency of the carrier. The discrete symbol sequence *s[m]* comprises of an alphabet distinctive for the modulation type. The waveform *g(t)* is a real valued signal pulse whose shape influences the spectrum of the modulated signal.

1.3.1 Aplitude Shift Keying (ASK)

ASK, also known as pulse amplitude modulation, is the simplest digital modulation scheme. The alphabet consists of M=2b points in the real line of the signal space where each point represents a sequence of 'b' bits. Therefore the symbols are represented by different amplitude levels of the modulated signal, as shown in figure 1.6.

The instantaneous amplitude of the ASK modulated signal can be expressed as

$$a(t) = A_c|s(t)| \tag{1.9}$$

Where

$$s(t) = \sum_{m=-\infty}^{\infty} s[m]g(t - mT) \tag{1.10}$$

i.e. the absolute value of the symbol function $s(t)$ with different amplitude

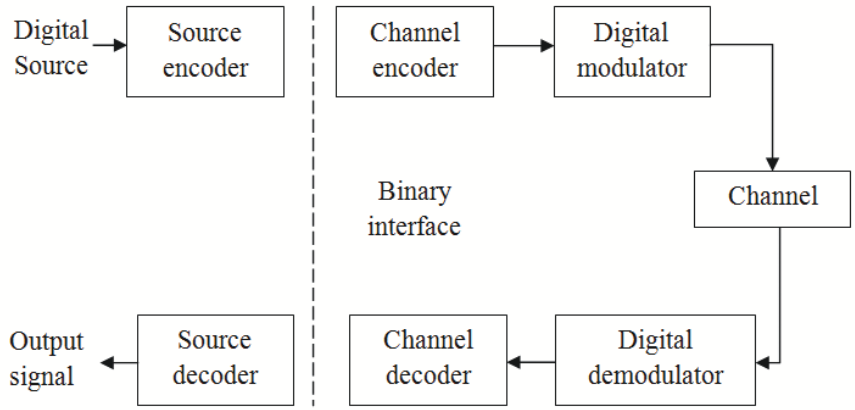

Figure 1.5: Basic block diagram of the digital communication system

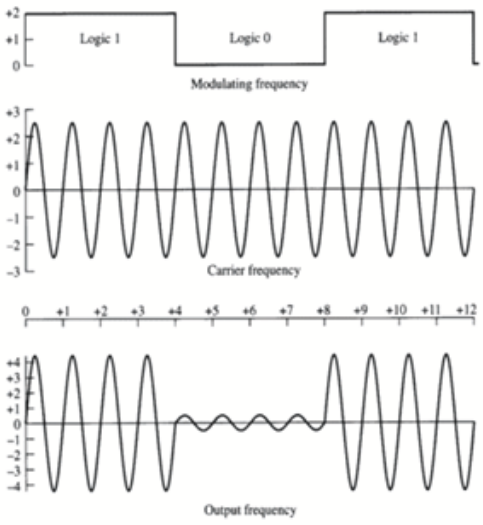

Figure 1.6: Signal waveform of the Amplitude Shift Keying modulation

levels scaled by A_c. The instantaneous phase is obtained by

$$\phi(t) = u\left(-s(t)\right)\pi + w_c t \tag{1.11}$$

Where u(t) is the unit step function. The instantaneous frequency may be expressed as

$$f(t) = -\frac{s(t)}{2}\delta\left(-s(t)\right) + f_c \tag{1.12}$$

1.3.2 Phase Shift Keying (PSK)

Phase shift keying is obtained by defining a unique phase state of the carrier for every symbol, as shown in figure 1.7. The symbols do not have any effect in the instantaneous amplitude.

The analytic PSK modulated signal may be expressed as

$$z(t) = A_c \sum_{m=-\infty}^{\infty} e^{j(w_c t + \phi[m])} g(t - mT) \tag{1.13}$$

The instantaneous phase is

$$\phi(t) = \omega_c t + \sum_{m=-\infty}^{\infty} \phi[m]\left[u\left(t - \frac{m-1}{2}T\right) - u\left(t - \frac{m+1}{2}T\right)\right] \tag{1.14}$$

where the unit step functions pick up the correct phase term in every time instant. The phase of the modulated signal consists of the phase states caused by the symbol sequence. The instantaneous frequency is obtained by

$$f(t) = f_c + \frac{1}{2\pi} \sum_{m=-\infty}^{\infty} \phi[m] \left[\delta \left(t - \frac{m-1}{2} T \right) - \delta \left(t - \frac{m+1}{2} T \right) \right] \tag{1.15}$$

PSK is a constant energy modulation scheme. If we want to transmit more bits in one symbol, then we simply increase the number of points in the constellation. If we want to transmit 3 bits per symbol then we divide the constellation in 8 divisions. For 4 bits, we have to divide the constellation angles into 16 points. But increasing the number of points will bring each point closer to its neighboring points and when the channel introduces noise the points will shift places by certain degrees. The demodulator then will not be able to recognize and separate such points due to the low margin of spacing, hence increasing the probability of error.

1.3.3 Frequency Shift Keying (FSK)

The FSK modulated signal comprises of pulses having different frequencies depending on the symbol, as shown in figure 1.8. The phase of the FSK signal can be continuous or discontinuous depending on the duration of the pulses.

The analytic FSK modulated signal may be expressed as follows

$$z(t) = A_c exp \left\{ j \left[\omega_c t + \omega_\Delta \int^t s(\tau) d\tau \right] \right\} \tag{1.16}$$

Where $s(t)$ is same as in 1.12. w_Δ is the frequency difference of two adjacent pulses. The envelope of the FSK signal is constant. The band-width of the

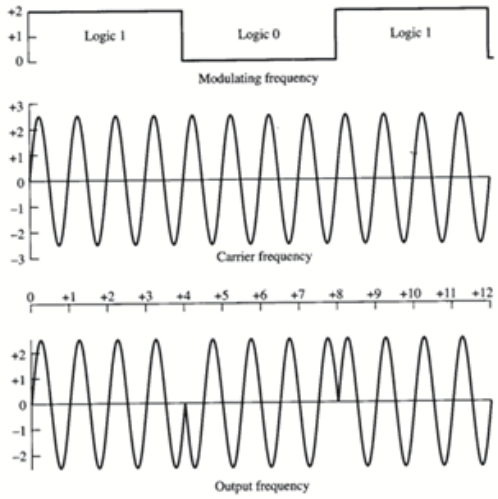

Figure 1.7: Signal waveform of the Phase Shift Keying modulation

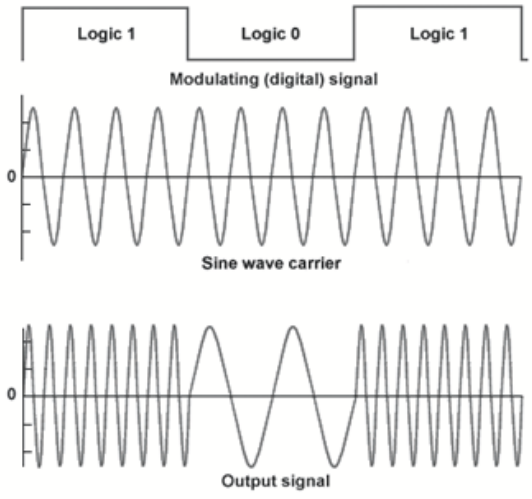

Figure 1.8: Signal waveform of the Frequency Shift Keying modulation

FSK signal may be reduced by choosing $f_\Delta = 1/(2T)$ which is called minimum shift keying MSK. By choosing *g(t)* as low pass filter with Gaussian shape, we get Gaussian MSK which is used in global system for mobiles (GSM) [5]. The instantaneous phase of FSK is given by

$$\phi(t) = \omega_c t exp\left\{ j\left[\omega_c t + \omega_\Delta \int^t s(\tau) d\tau \right] \right\} \tag{1.17}$$

The instantaneous frequency then becomes

$$f(t) = f_c + f_\Delta s(t) \tag{1.18}$$

i.e. the instantaneous frequency varies with respect to the symbol values.

1.4 Communication in Military / Software Radio

On the demodulation part of the communication system the signal modulation scheme is required to be remapped to its analog domain, separated from the corruptions induced by the channel. This might not be the case in military and intelligence applications such as surveillance, monitoring and other covert eaves drop applications. The received signal waveform is as given

$$f(t) = s_m + n(t) \tag{1.19}$$

The received signal $s_m(t)$ is distorted by stationary Gaussian white noise usually uncorrelated with the signal $s_m(t)$. More importantly, the modulation scheme of $s_m(t)$ is unknown.

The Software Radio and Cognitive Radio are the practical applications, for example in a wireless network environment, where it is required for an unknown incoming signal to be routed to the right demodulator. The advent of realizable SR allows the implementation of creative transceiver designs, which can dynamically adapt to the communications channel and user applications [6].

The automatic selection of the correct modulation type is a major advantage in a cognitive wireless network. The modulation schemes can be switched according to the changing channel capacity. This increases or decreases the baud rate, resulting in effective use of channel capacity and reduced bit error rate (BER). A generic overview of a software based demodulator is illustrated in figure 1.9.

1.5 Pattern Recognition of Communication Signals

We find computer based pattern classification important in many areas and applications, such as fault detection, medical diagnosis, biometric identification, speech and optical character recognition and, as we will focus on, communication signal recognition. Despite this myriad types of data (e.g. sensor data, digital images, acoustic or radio signals). The term pattern recognition, machine learning,

artificial intelligence or computational neurosciences are all very closely related terminologies, see figure 1.10. They all are branches of the scientific discipline that learn to categorize instances or objects into respective categories or otherwise mentioned as classes.

There exists a generic pattern classification approach as shown in figure 1.11. The figure shows that first some form of pre-processing is performed. The preprocessing stage include tasks such as power equalization, reduction of noise, carrier frequency estimation etc. The Basic pre-processing requirements for modulation classification are to obtain a signal representation that is consistent and to reduce degradations that may confuse the classifier. The preprocessing stage also converts the signal to baseband and filters it. Next is the process of reducing the dimensions of the data by extracting a smaller number of features that emphasize the characteristics of, and distinctions between, the classes. The features then serve as inputs to the classifier, which performs the final classification task. In order for the classifier to perform classification, it must obtain information on what data represents what class. This is often referred to as learning. That is, based on pre-labeled examples from each class (training examples) the classifier learns to map the data to their corresponding class. It also learns to generalize the data such that new and unclassified examples are classified accordingly.
Having obtained a feature vector a classifier must then use that information to output a class label or a set of probabilities / confidence levels that indicate the prediction results. There is no classifier type that is superior to the others for all

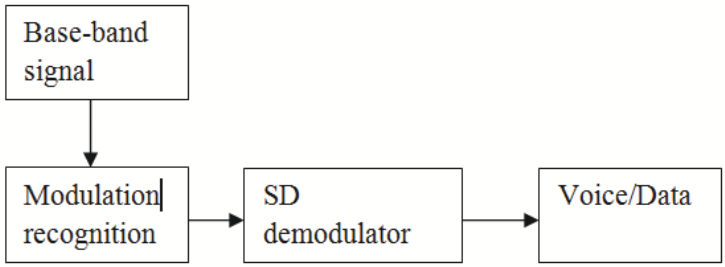

Figure 1.9: Placement of the modulation recognizer in the receiver architecture

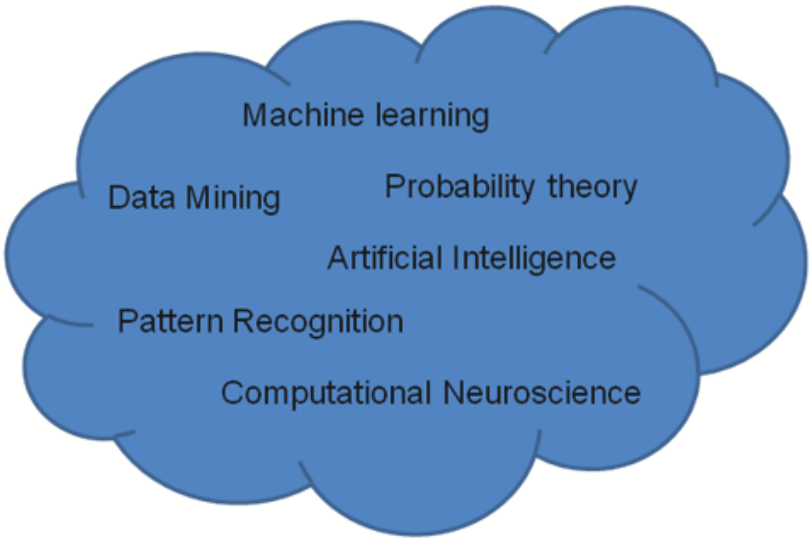

Figure 1.10: Different terminologies related to pattern recognition

types of classification problems.

Take for example the problem of classifying speech into either male of female voice. The distinctive features would be the different frequency components. Usually men speak at a lower frequency than women which gives them the lower pitch and hence the heavier voice. A suitable classifier can be chosen to use pitch, frequency and standard deviation in classification. Figure 1.12 shows an example plot of the frequency vs the standard deviation between the male and female voice. Each 'o' corresponds to a female voice sample where an 'x' represents a male voice sample. A simple classifier would create a hypothetical line between the two classes. Such division will serve as a classification scheme to identify the two classes of objects. This straight line is known as a decision threshold.

In our particular case we need to classify objects such as signal waveforms into popular modulation types such as the amplitude shift keying, phase shift keying or frequency shift keying.

1.5.1 Types of modulation classifiers:

There are primarily two types of techniques for classification, parametric and non-parametric. The parametric modulation classifier starts off with a known distribution model (e.g. Uniform, Gaussian) and the objective is to find suitable feature values from the training data. These are used to estimate the actual distribution. The non-parametric modulation classifiers focus on modeling distributions from the training data only. They are not limited by the standard distributions. Thus

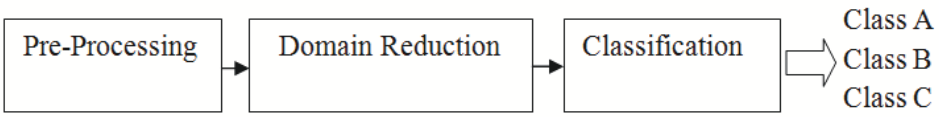

Figure 1.11: Block diagram of a generalized classifier

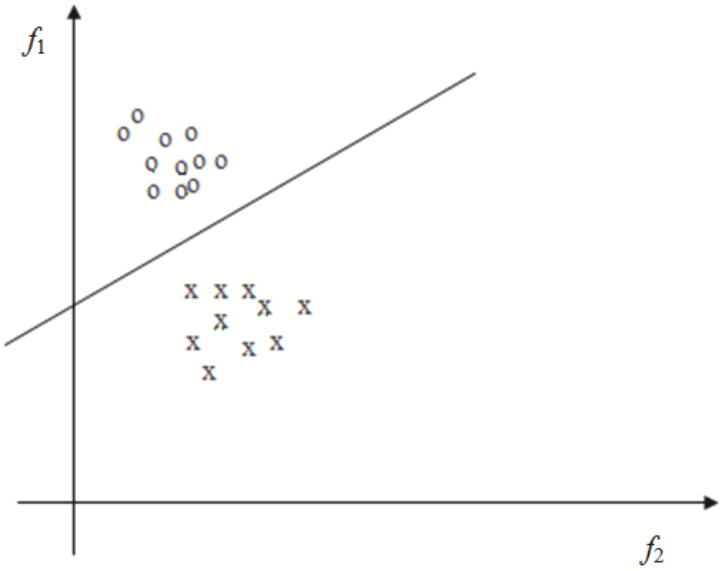

Figure 1.12: frequency vs standard deviation plot for gender classification using speech

it can provide wider range of distribution models with greater number of features. The non parametric technique has been proposed for this modulation classifier. Further sub-divisions of the non-parametric technique consist of 'Sequential' and 'non-sequential' classifiers. The details of these type of classifiers are as follows.

Sequential classifiers

The classifiers with variable number of features are called sequential classifiers. This type of classifier starts the process of classification with evaluation of one feature only. If the process is decidable, classification process is finished. On the other hand, if the classifier is unable to predict the class, then another feature is added and the process is repeated till the class is predicted.

Non-Sequential classifiers

The classifiers with fixed number of features are called non-sequential classifiers. These classifiers have signal characteristic with an exact number of features. This number is fixed during the whole classification process. The proposed modulation classification algorithm uses non-sequential classifier as the main identification method.

1.6 Classification of Digital Modulation Schemes

A digital automatic modulation identifier (AMI) is a device that automatically recognizes the modulation type of received radio signals. An AMI finds its uses in military and civilian communications applications including signal confirmation, interference identification, spectrum monitoring, signal surveillance, electronic warfare, and military threat analysis. When the modulation schemes of a received signal are identified, an appropriate demodulator can be selected to recover the information. Methods used for modulation classification are implemented as intermediate stages between received signals and the decryption or data recovery stage, see Figure 1.13. The modulation classification block shown in figure 1.13 is illustrated with examples in figure 1.14. The process of AMI can be used for analog as well as digital modulations.

Modulation classification is a complex task, especially in multi-signal environments, such as HF band, where many current civilian and military radio communication systems operate. It is very difficult to analyze and identify differently modulated signals in real time due to the high signal density and fading effects in HF band. Also, signal degradation factors such as multipath propagation, frequency-selective fading, and time-varying channel add to the complexity of the task. Also, there might be little to no a-priori information provided about the incoming signal.

The first attempts at developing modulation signal classification appeared in

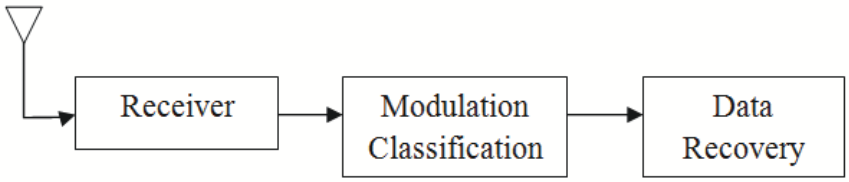

Figure 1.13: Placement of the modulation classification at the receiver

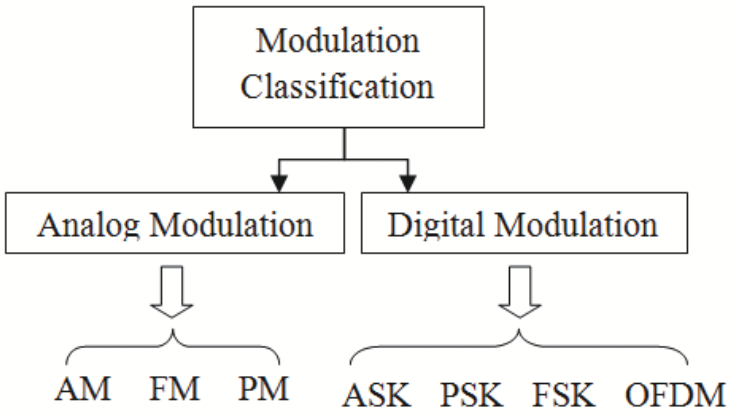

Figure 1.14: Different divisions of modulation classification

[7-10]. More recently, Azzouz and Nandi introduced a decision theoretic approach in [11]. Six signal features were used in their work for both analog and digital modulation signals and a number of user-defined thresholds were employed to perform classification. A total of 13 classes were considered with an average classification rate of 93% at 15 dB signal-to-noise ratio (SNR). Nandi and Azzouz extended their work by using artificial neural networks for both modulation signal types in [12], where there was no need to specify a threshold manually.

1.7 Blind / semi blind classification techniques

In the previously defined example of classifying male and female voice signals, a training data set was provided to the classifier. The provided data was used critically in the identification of further instances. This pre-classification data is called a-priori information, and the practice of using a-priori information is known as 'Supervised Learning'. The supervised classification method requires the presence of training data set typically defined by the expert. The term signal classification implies that there exists some a-priori knowledge about the various types of communication signals that are to be analyzed [13].

In contrast to supervised learning, blind detection have no a-priori information of the test instance. To classify blindly, no a-priori information is used from the test instance, but certain features are extracted from a sample instance that correlates to the test instance. An example is the classification of modulation schemes, which is performed in a non-cooperative environment. One cannot assume full a-priori

knowledge, such as the carrier frequency, bit-rate, phase offset, signal power and symbol length etc. This is partly due to noise interference and channel effects present in the communication channel. Blind detection process suffers from the complexity of carefully selecting the most effective features that will work on a large number of test instances. Also, the selection of classifier has great impact on the classification outcome.

The third classification method, used in the proposed classifier, is semi-blind detection. Semi-blind detection uses some small amount of a-priori information of the test instances. Such as, knowledge of the range of carrier frequencies or the range of bit-rates might be required. Semi-blind detection will yield more accurate results with lower number of computations as compared to the blind detection. This is because the method has a limited number of possible test patterns to classify.

1.8 Problem Statement

Much of the existing work on modulation classification has been overly focused on reporting classification success rates that exceed that of others by tweaking classifier parameters. Not many classification approaches have been proven to work reliably with signals that have low SNR (below 5dB), on a broad range of classification schemes [37-42].

It is also uncommon to find a modulation classifier that performs well on low SNR and can also be realized into a compact hardware module [68, 69, 75, 80].

A tradeoff exists between the classification accuracy and architecture complexity of the hardware implemented modulation classifier. Thus, a real-time modulation classification system is highly sought after.

Lastly, a modulation classifier cannot assume full a-priori knowledge of signals in a non-cooperative environment. Also, the classifier should be able to handle high degree of noise and channel degradations.

Hence, the need is to develop an automatic modulation classifier that

- Performs with high accuracy in low SNR conditions
- Performs classification using small quantities of sampled data
- Performs classification in real time
- Identifies several modulation schemes
- Adapts to channel effects like path loss and multipath
- Should not depend on full a-priori knowledge of the signal
- Should be practical and low in complexity

1.9 Objectives of the thesis

The proposed work aims at developing a robust approach for modulation classification in identifying digitally modulated signals. The classification is carried out to identify modulation schemes; 2-ASK, 4-ASK, 2-PSK, 4-PSK, 2-FSK, and 4-FSK. The modulation types were restricted to the types commonly used in radio

communication. The proposed model is aimed to be low in complexity and computationally in-expensive. Also, the developed modulation classification method is evaluated under degrading channel conditions such as noise and multipath effects. Finally, the proposed modulation classifier was implemented on testbed hardware for evaluation of real time performance.

The fundamental concept behind the proposed work is the use of pattern recognition techniques using a Bayesian approach. The proposed method uses the first four moments namely the mean, variance, skewness and kurtosis as the features. These attributes are fed to the Bayesian classifier as training data. The features are selected from statistical moments, because moments can accurately characterize the probability density function (PDF) of a signal [65].

The proposed modulation classifier is implemented on a field programmable gate array (FPGA) platform. A detailed rationale, on selecting a suitable platform, is provided in chapter 3. A graphical user interface (GUI) is developed to interface the FPGA with a personal computer (PC). The GUI provides an interactive method to record the performance of the proposed modulation classifier.

CHAPTER 2

LITERATURE REVIEW

In this chapter, the different approaches used in modulation classification are reviewed. The set of features used in the classification process are reviewed and studied. Also the techniques used for real time implementation are discussed.

2.1 Main Approaches used for Modulation Classification

In general, the task of automatic modulation classification is divided into two groups; a decision theoretic approach and a pattern recognition approach, as depicted in figure 2.1.

The decision theoretic approach, in some literature also called likelihood-based (LB) approach [14-29], is based on composite hypothesis testing. In this method a few signal parameters are used to process the likelihood ratio and then compared

to an evaluated threshold.

In the pattern recognition method, classification is achieved by using several behavioral attributes or features [31-72]. Several features are estimated and a decision is made by using a suitable classifier. A feature based classifier may not be highly accurate but it is simpler to implement. A near-optimal performance can be achieved by properly designing the method, pertinent to the application. Thus the performance of this model relies on defining a set of features that pair up optimally to the designated classifier.

2.1.1 Likelihood-based (LB) approach

Likelihood based (LB) classifier works on the basis of a likelihood function (LF). As the name describes, this function employs a mathematical function that predicts the likeliness of the signal to fall in a certain category. The measure of a function being categorized is quantified against a threshold value that is related to the likelihood function. This is a challenging task when identifying many different modulation schemes.

In the LB approach, the classification is viewed as a multiple hypothesis testing problem, where a hypothesis, H_i is arbitrarily assigned to the ith modulation type of m possible types. The LB classification is based on the conditional PDF $p(x|Hi)$, i=1, ...,m, where x is the observation; e.g. a sampled phase component. If the observation sequence $X[k]$, k=1, ...,n, is independent and identically distributed (i.i.d), the likelihood function (LF), $L(x|Hi)$, can be expressed as

$$p(x|H_i) = \pi_{l=1}^{n} p(X[k]|H_i) \triangleq L(x|H_i) \tag{2.1}$$

The block diagram of a general maximum likelihood classifier is shown in figure 2.2.

Likelihood based classifiers are generally quite accurate but the high probability of correct classification comes at a price of computational complexity.

Wei and Mendel developed a maximum likelihood method for classification of digital amplitude-phase modulations in [14]. The method is applicable to any constellation-based modulation types in an additive white Gaussian noise (AWGN) environment. The authors obtained the theoretical performance of the ML modulation classifier that works under ideal conditions, and can serve as an upper bound of performance for any classifier. All signal parameters were assumed to be available.

Boiteau and Le-Martret proposed a general maximum likelihood classifier (GMLC) in [15] based on an approximation of the likelihood function. The authors derived equations of GMLC in the case of linear modulation and applied them to the MPSK/M'PSK classification. They deduced the likelihood function of an observation by the measure of correlation between the empirical and the spectral higher-order statistics. The GMLC provides a theoretical foundation for many empirical classification systems including systems that exploit the cyclostationary property of the modulated signals.

Yang and Liu proposed an asymptotic optimal algorithm for classifying the mod-

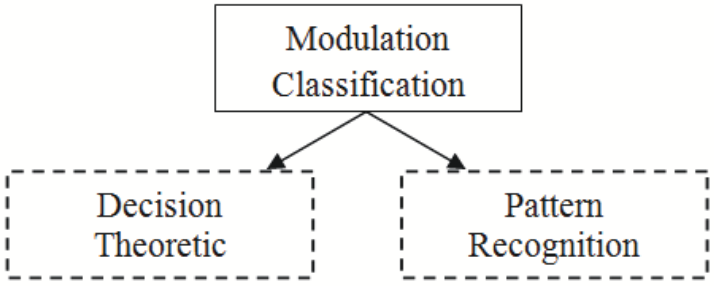

Figure 2.1: Basic approaches used in modulation classification

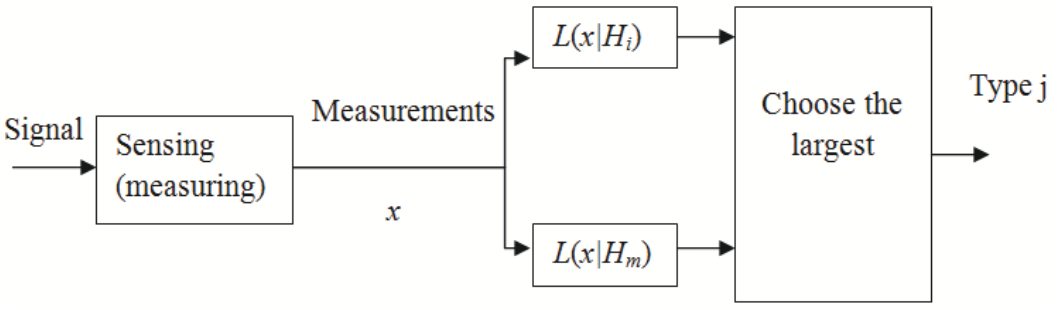

Figure 2.2: Block diagram of the Maximum Likelihood classification

ulation type of general MPSK signals in [16]. Yang published the same results earlier with Soliman in [17] and in [18] with slightly different test statistics. The authors developed the classification algorithm by employing the exact phase distribution of a received MPSK signal, which was expressed in terms of the Fourier series expansion. The authors showed a structure of this proposed classifier for CW, BPSK, QPSK, and 8PSK. The maximum a posteriori (MAP) probability criterion was used to develop a multiple hypothesis classification rule. The MAP criterion reduced eventually to a ML classifier because the hypotheses were assumed equally likely. The performance of this classifier was shown to be more effective than the earlier classifiers in [16, 18]. The SNR was assumed to be known.

Quasi Log-Likelihood Ratio Classifier

Kim and Polydoros proposed a quasi log-likelihood ratio classifier (qLLRC) for BPSK and QPSK in [19, 20]. They compared it to the more traditional, ad-hoc techniques of the square-law classifier (SLC) and the phase-based classifier (PBC). The qLLRC was derived by approximating the likelihood functions of the phase modulated digital signals in white Gaussian noise. The authors showed that its performance is significantly better than that of the intuitively designed PBC, or the conventional SLC. The capability of the qLLRC algorithm is limited because it is only valid for low SNR (SNR¡ 0dB). All signal parameters such as the carrier frequency, initial phase, symbol rate and SNR were assumed to be available. The classifier can only be used to discriminate between BPSK and QPSK signals.

M th-law Classifier versus qM-Rule

Hwang and Polydoros proposed a maximum likelihood classifier based on the likelihood function of MPSK and M-QAM signals in additive white Gaussian noise in [21]. The authors derived simplified versions of the LF for each modulation type denoted by the qM-statistics. The qM-classifier can be interpreted as a synchronous, pulse-shape matched filter. The correct classification probability was approximated in low SNR (SNR $\langle$ 0dB) with long observation time, i.e., N $\gg$ 1 symbols. To achieve the same performance with the M th-law classifier, additional gain in the SNR was shown to be more than 2dB. The qM-rule is only valid for low SNR (SNR $\langle$ 0dB) and all signal parameters such as the carrier frequency, initial phase, symbol rate and SNR were assumed to be available.

Average Log-Likelihood Ratio Classifier

Long, Chugg and Polydoros extended the low SNR methods to moderate and high SNR environments in [22]. The authors presented the Qm-rule based on the average log-likelihood ratio (ALLR) and an approximate expression for the pdf of the Qm-statistic was developed for medium and high SNR cases. The performance of the Qm-rule was evaluated in four different environments; e.g. in TV and CPFSK interference. The approximation of the ALLR demonstrated an ability to dramatically improve the performance compared to the qM-rule in [21]. The classifier was developed for binary hypothesis testing and all signal parameters were assumed to be available.

Average Likelihood Ratio Test (ALRT)

The average likelihood ratio test assumes the probability density function of the test vector and considers them as random variables. ALRT produces better results than the remaining likelihood ratio tests but it requires explicit knowledge of the signal power and noise variance for the channel. It was previously described that the likelihood based classifiers work on a function for prediction, this function under the ALRT hypothesis is given by

$$\Lambda_A^{(i)}[r(t)]) = \int \Lambda[r(t)|v_i, H_i]\, p(v_i|H_i) dv_i \tag{2.2}$$

where $\Lambda[r(t)|v_i, H_i]$ is the conditional LB function of the received signal, and v_i is the unknown vector. H_i represents the i^{th} modulation and $p(v_i|H_i)$ represents the a priori probability v_i given H_i.

Sills [23] attempted to classify linear modulations with additive white Gaussian noise (AWGN). This method uses ALRT to classify the modulation type without the a-priori knowledge of the carrier phase.

Hong and Ho [24] identified BPSK and QPSK signals with unknown channel amplitude α. The Bayes test is derived by averaging over the Rayleigh distributed random variable α. The symbol time T_s, the carrier frequency f_c, the carrier phase ϕ_c, the power spectral density (PSD) of AWGN channel N_O, the nominal power level S and the variance of α are presumed to be known to the receiver.

As previously discussed about the high computational complexity of the likelihood

based classifiers, ALRT algorithm is the most expensive type in LB classifiers. To overcome the high computations of ALRT two sub-optimal LB classifier GLRT and HLRT algorithms are used.

Generalized Likelihood Ratio Test (GLRT)

The generalized likelihood ratio test is similar to the ALRT except that in GLRT does not require the knowledge of noise power to compute the likelihood function. The generalized likelihood ratio test assumes the signal vector as a deterministic value.

In [25] the thresholds were set using the histogram method and GLRT tests. GLRT has a few advantages over the remaining two types, such that there is no computation of exponential functions and thus it is a good candidate for implementation.

Hybrid Likelihood Ratio Test (GLRT)

As the name suggests, the hybrid likelihood ratio test is a combination of ALRT and GLRT. In HLRT some of the unknown parameters are treated as deterministic and some as random variables. The likelihood function is given by

$$\Lambda_G^{(i)}[r(t)] = max_{v_i} \int \lambda[r(t)|v_i, v_{i_2}, H_i]p(v_{i_2}|H_i)dv_{i_2} \tag{2.3}$$

where $v_i = [v_{i_1}^\dagger v_{i_2}^\dagger]^\dagger$ and, v_{i_1} represents the signal parameters and it is characterized as a vector of deterministic type. Whereas v_{i_2} shows the data symbol

vector and it is characterized as random variable vector.

GLRT and HLRT methods were used in an ISI and additive white noise environment [26, 27]. The likelihood function was calculated using the Maximum Likelihood estimation of the signal vector and channel coefficients. Thresholds were set by trial and error method.

Vito et al. [28] presents two classification methods. They worked on direct acquisition devices such as a waveform digitizer. The modular approach makes it easy to add or remove modulation identification steps from the overall method. This technique makes the process of identification adaptable to many different types of modulation schemes.

Another approach in [29] uses the likelihood test for modulation classi?cation. The method uses a look up table (LUT) to identify unknown symbols from the signal. This method uses the pre evaluated test functions for the likelihood ratio test. These values are matched to the LUT and modulation schemes are classified in real time. The taxing multiplications are reduced by additions and exponentials involved in the traditional likelihood function resulting in only 2 to 3 additions to get the memory address index.

2.1.2 Feature based approach

The feature based classifiers follow a different approach, such that they extract features from the signal and a decision is taken by using a suitable classifier [30]. The feature based method is simpler as compared to the likelihood based classifier

but is less accurate. The decision of choosing between the two classifiers is primarily based on the application. There lays no standard guideline to determine which features should be taken to ensure high probability of correct classification. A more detailed description of the feature based classification approach is discussed in chapter 3.

Following are the common type of features or domains from which distinguishing factors are extracted for identification

- Signal amplitude, phase and frequency variances [31]
- Zero crossing interval [32,33]
- Signal wavelet transform [34,35]
- Moments [36-38]
- Cumulants and cyclic cumulants [39-50]
- Signal entropy [51-52]
- Constellation shape [53]

After the feature extraction following are the common type of decision making techniques used in the modulation classification process [54].

- PDF based [39-41]
- Hellinger distance [55-56]
- Euclidian distance [45-50]

- Clustering method [57-58]

We will discuss in what follows some of the most important features used in modulation classification. Following are the categorized descriptions and literature review of the afore-mentioned features.

Amplitude, Phase, Frequency Features

When analyzing a signal waveform, the most out-standing parameter of the waveform is its amplitude, phase and frequency. Just by visually comparing the waveforms of ASK, PSK and FSK one can identify several substantial differences in them. For e.g. FSK and PSK have constant amplitude as compared to ASK. Likewise, PSK have phase information that ASK does not have. Hence PSK and ASK can be compared against this parameter.

The zero-crossing (ZC) method was used to differentiate between FSK and PSK in [32] and [33]. The zero crossing count is the measure of zero amplitude passes in a given sample data space. Zero crossing can be translated into a measure of relevant frequency. PSK and FSK are differentiated by the evaluated variance.

Similar technique was used in [59-60] by using the instantaneous frequency as a feature. The variance of signal frequency was used to distinguish between FSK and PSK. This variance, for each given modulation scheme takes on a unique range of values. This range is set apart by comparing the two variances against a threshold, and classifying the modulation scheme.

In [12, 31] the instantaneous amplitude was used to identify between BASK and QASK modulations. The difference in the variance of instantaneous amplitude is

higher in QASK then in BASK. A decision tree classifier was used to compare the features against a threshold.

In [16] the probability density function of the phase was analyzed for M-ary PSK identification. PDF of the phase gives out information of the order of Phase shift keying modulations. An approximate parameter can be derived by using Tick-honov PDF and Fourier series to estimate PSK probability [16]. A likelihood ratio test was used for comparing the estimated probability and making the decision.

In [36-38] the PDF of moments were employed for decision making upon the moments of the phase. The moments were assumed to have Gaussian distributions. The decision stage of the classification was further improved by comparing the moments against a threshold. In [61] the order of PSK was identified by comparing the phase difference between two neighboring symbols. The decision error was reduced by generating a histogram of the difference and then comparing it against a pre-determined pattern. The characteristic function of phase was used to identify order of PSK in [62]. This method was extended to include QAM modulation identification by also inducting the amplitude in the analysis [63]. Similar work with PSK and QAM was done in [10] by using kurtosis of amplitude as the primary feature. Kurtosis of a distribution shows that how much the data is peaked.

Wavelet transform based features

Wavelet transform is an alternate of the classical Fourier transform (FT) that has the advantage of being faster than the FT. Wavelet transforms can be used to

analyze the frequency domain and envelope of a signal. The wavelet transform captures the characteristics of the received signal in time and frequency by producing scaled and shifted derivations of the original signal wavelet.

The wavelet transform serves as a productive method to set aside the subtle changes in the amplitude, phase and frequency of a signal. The Haar wavelet is the simplest in the wavelet family. The Haar wavelet uses square shaped functions to form the wavelet basis. In [34-35] the Haar wavelet transform (HWT) differentiates PSK, FSK and QAM. HWT will be a constant value for PSK while in the case of FSK and QAM the HWT becomes a staircase function. In [34] the HWT variance with no amplitude normalization was used to set apart FSK from PSK and QAM. In the second stage the same variance was normalized to identify between PSK and QAM. The decision also included a threshold matching.

It was studied in [35] that different modes of PSK produce different number of peaks in the Haar wavelet transform. These peaks were analyzed to identify the order of PSK. Histograms were drawn and compared to the PDF for decision making.

A combined technique of wavelet transforms and Artificial Neural Networks was used in [64]. To obtain the features from the signal, Daubechies wavelet was used. The Daubechies wavelet has an advantage of disseminating the signal making it easier to gather attributes. A Radial Basis Function Neural Network (RBFNN) was used at the decision making stage.

Signal statistics-based algorithms

Signal statistics can be used to characterize the PDF and to estimate fX(x) from experimental measurements. The statistics can be determined analytically via the characteristic function of X. The characteristic function is given by

$$\Phi_X(\omega) \triangleq E\left[e^{j\omega X}\right] = \int_{-\infty}^{\infty} f_X(x) e^{j\omega x} dx \tag{2.4}$$

If we take the natural logarithm of the characteristic function, we get the cumulative function $\Psi_Z(\omega)$ or the second characteristic function. In [39] cumulant based features were used to differentiate between ASK, PSK and QAM. A likelihood based method using the PDF of the cumulants was used for decision making. Moments are another quantitative measure of the statistical distribution. It was shown in [65] that the knowledge of all moments of a data set can completely characterize the PDF.

The 2^{nd} and 6^{th} order statistical moments were used in [40] as signal features. The classifier discriminates between PSK and QAM. In [66] multiple moments and cumulants were joined for a neural network based classifier to distinguish between PSK, FSK and QAM.

In [67] the authors use pattern recognition approach to classify modulations in frequency selective fading channels. The features were selected from moments and higher order cumulants. A radial basis function neural network (RBFNN) was used as the classifier. Neural network classifiers work on the basis of biological neurons. A decision matrix is developed by evaluating weighted neurological

paths.

A set of linear modulations and QAM were classified in fading environment using the blind alphabet equalization algorithm [41]. The extracted features were cumulant based and amplitude, phase and frequency based features.

Moments were used to classify between QPSK and 16-QAM in [42]. The $E[X^4]$ and the $(E[X^2])^2$ moments were used as features. Their linear combination and correlation was utilized to evaluate the coefficients and the delay vector for correct classification.

Signal cyclostationarity is another commonly adopted technique for classification [46-50]. The cyclic cumulants of multiple orders are seen to be used as features to classify modulation types.

An algorithm was developed in [46] to make use of the signal cyclostationarity for modulation classification. The cyclic-cumulants were selected as the features upto the nth order. This NP hard algorithm is not feasible for practical implementation. Certain sub-optimal solutions to the algorithm were proposed using the Euclidian distance at the decision stage.

In [68] the cycle frequency domain pro?le (CDP) was employed for signal classification. The threshold test method was used to obtain the features. A novel feature called the crest factor was obtained. At the decision stage a Hidden Markov Model classifier was used.

It was shown in [47] that single and multiple signals at the receiver can also be identified by using joint detection classification. Cyclic frequencies have the prop-

erty of selecting and distinguishing different symbol rates. All signal parameters related to the bandwidth, symbol rate and amplitude were calculated. In [48-50] the minimum Euclidian distance was calculated between the signal vector and the feature vectors for the modulation classification.

Grimaldi [69] proposed a method to distinguish between single carrier and multiple carrier waveforms. They have used normality tests and decision tree based structure to set apart single-carrier and multi-carrier modulation types. Multi carrier modulations are just random variables that are independent and identically distributed, hence based on the central limit theorem the amplitude of multi-carrier modulations follow a Gaussian distribution. The Giannakis-Tsatsanis test was used as the normality test in the decision tree classifier.

The most commonly classified modulation types are ASK, PSK, FSK and QAM due to their usage in common communication applications. Other modulation types were also analyzed for example MSK [70], OQPSK [71] and CPFSK [72].

2.2 Modulation Classifiers Implemented on Hardware

The development of any technology is aimed to result in a practical real world application. The first stage to a handle a mathematical problem is to attain a rationale and an effective solution. If the solution is competitive enough as to outweigh other solutions then it becomes a probable candidate for realization and

practical use. Then we move to the stage of reducing the complexity of the developed solution and verifying it on testbed hardware.

In the case of modulation classification, much work has been done to develop accurate classifiers. The scientific community dealing with this problem is now moving to the stage of developing practical solutions. The requirement of a practical software radio for military communication was discussed and in [17]. The objective is to develop a reduced complexity classification algorithm that can be ported to multiple platforms while maintaining the accuracy.

Some of the common platforms for the implementation of modulation classifiers are

1. Digital Signal Processing kits
2. Field Programmable Gate Arrays
3. General Purpose Processors
4. Rapid Radio Devices

Following is a literature review of the previously implemented modulation classifiers. A more detailed discussion on the advantages and disadvantages of each type is given in chapter 3. Also, in chapter 3, a comparison is performed to select the most suitable platform for the proposed modulation classifier.

2.2.1 Digital Signal Processing (DSP) kits

DSP kits and DSP development boards are specially designed for testing signal processing methods on hardware platforms. DSP implementations offer fully deterministic function execution times and high code optimization options. But they lack the robustness required for mobile work. They have highly constrained memory capabilites for waveform storage and buffering.

A hierarchical neural network modulation classifier was implemented on a DSP kit in [73]. The TMS320C6701 was chosen as the processor of choice and multiple types of classifiers were tested on the processor. While classifying analog and digital modulations, 31 features were used to identify a total of 11 modulation types. Genetic algorithm was used to generate the features from the amplitude, phase and frequency of the waveform. The recorded performance specifications were 680 clock cycles per node and 0.4 milliseconds response time of the classification stage. Unlike many classifiers, [73] differentiates between single and multi carrier signals in order to also identify OFDM signals. The proposed method was implemented in C language on the Texas Instruments TMS320C6711 digital signal processing (DSP) starter kit.

In [74], the numerical simulations of the five types of modulation classifiers, DTC, MDC, NNC and SVC were performed. The prototype modulation classifier was implemented on TMS320C6203 as a reprogrammable software radio. Performance evaluations were done using field experiments. The classifier performed with 95In [75] a feature based classifier was implemented on TMS320C6201 as part of the

channelized receiver exploiting A/D converters, FPGAs and programmable DSPs. CORDIC arithmetic is utilized to dissemble many complicated arithmetic operations into addition and shift operations.

One method for signal classification and monitoring using a DSP module and a personal computer is described in [76]. In the process of modulation classification the spectrum generator was used with the digital signal processors. The DSP module contained a processor (TMS320C50), microcontroller (MC68HC1l), A/D converter, D/A converter, changeable and fixed memories as RAM and PROM respectively.

Enrico Buracchini [77] suggests a software replacement of the IF and baseband stage to improve the performance. This software architecture was also tested on DSP and GPP based hardware testbeds. It was concluded in [77] that DSP based testbeds provide better performance with low complexity circuits. A DSP testbed was used to implement a military JTRS and SDR model.

A DSP architecture for proposed in [78] to implement a WCDMA system. The architecture proposes a baseband solution for GSM, GPRS and 802.11b. All physical layer components were replaced by software architecture that was implemented on the DSP kit.

The proposed solution of the modulation classifier in [79] is based on feature based approach. The signal envelope and its instantaneous frequency are selected as features. There are 16 possible values of the classification time, found in the range 300 - 1050 milliseconds depending upon the selection of up to 16 features.

2.2.2 Field Programmable Gate Array (FPGA)

FPGA is a reconfigurable or reprogrammable hardware platform to develop and run programs and algorithms. FPGAs can achieve very high performance specifications and very fine low level programming.

A Matlab (Simulink) based classifier is proposed in [80]. The target FPGA device was Xilinx Virtex v3200efg1156. The required power by the designed modules was 466mW. The propagation delay was calculated to be 70.396 nano-seconds, out of which 32.344ns was for logic and 38.052ns for routing. This translates into 45.9% and 54.1% for combinational logic and data path routes respectively. Synthesis reports depict the area utilization to be 32.11%. The total memory required is 714620 kilobytes.

A few related FPGA based implementations include Srikanteswara et al., [81]. They proposed a generalized study of reconfigurable computers for software radio. Hartenstein [82] developed coarse grain reconfigurable architectures. The focus was to introduce hardware accelerators in the software radio system. The control of the proposed accelerators was done by DSP or RISC. Cummings and Haruyama [83] used SRAM FPGA architectures. It was concluded in [83] that FPGAs perform faster than DSP boards in modular design flow. They can also provide parallel computing support with low power consumption.

The implementation of M-ary FSK and BPSK demodulators was reported in [84]. The TMS320C6203 board was used excluding the analog circuitry. The power spectral density (PSD) was used to classify M-ary FSK with BPSK.

John Huie et al [85] combined a FM modulator and demodulator into one software processing core. The software architecture was proposed for FPGA based implementation.

2.2.3 General Purpose Processors (GPP)

As the name implies, general purpose processors can be from a broad range of processors designed for personal computing. A GPP offer large amounts of program and user memory, floating point operation, better high level language development environments and are less expensive compared to dedicated DSPs.

A combined classifier is proposed [86] that identify both trained and un-trained modulation types. The classifier is based on two different neural network classifiers. The performance was evaluated on both semi-real and simulated data.

General purpose processors were used in [87] to develop a real time SDR. The classification algorithm was implemented on a 700MHz processor. The classifier is developed on the classification method of waveform space partitions and by using Hellinger distance approach. Possible applications include Wi-Fi, where the algorithm dynamics change according to the environment.

Zhao [88] presents an AMC (automatic modulation classifier) using an orthogonal wavelet-basis neural network approximator. The probability of correct classification was 98.61% at 8dB SNR. Real world signals were used in the process of identification with the wavelet analysis based feature extraction method. At the classification stage a BP neural network classifier was executed using Matlab based

interface.

2.2.4 Rapid Radio devices

Rapid Radio device is a generalized term for any hardware platform that can be customized for rapid code implementation. If the speed of deployment is paramount, and the prototyping platform has an abundance of resources, then the design process can be altered in a way to produce an operational yet possibly suboptimal receiver in a short time.

The prototype of an automatic digital modulation classifier is presented in [89]. A hierarchical architecture was proposed for hardware implementation. The complete testbed consists of an evaluation board with embedded Analog/Digital converter (ADC), frame grabber and a personal computer. The prototype hardware is composed of a Personal Computer (PC) with the framework realized in C++ language.

A rapid radio development framework was proposed in [90] that has easy to follow developed interfaces that guide the user in modulation classifier development and the implementation on hardware. Classification accuracy is, to no surprise, quite low.

In [91] a method for modulation classification is proposed together with a technique to implement the method using rapid framework. The framework is divided into stages where the system extracts certain parameters and then implements the demodulator and the proposed classification method on FPGA. The proposed

classifier is based on Bayesian network classification technique. The framework uses Matlab and Xilinx Coregen to develop module for the FPGA.

CHAPTER 3

THE PROPOSED MODULATION CLASSIFICATION FRAMEWORK

In a communication system, a message signal $m(t)$ is modulated and transmitted. This transmitted signal is received at the receiver where a classifier determines the modulation scheme of the received signal $r(t)$. This section talks about the basic framework developed for the modulation classifier. Also, the motivation and methodology that lead to the final developed algorithm has been discussed.

3.1 Basic digital modulation classifier framework

In general, the task of automatic modulation classification is divided into two groups; a decision theoretic approach and a pattern recognition approach as shown in figure 3.1.

The decision theoretic approach, also known as the likelihood-based approach is based on composite hypothesis testing. In this method a few signal parameters are used to process the likelihood ratio and then compared to an evaluated threshold.

In the pattern recognition method, classification is achieved by means of behavioral attributes or features. Several features are extracted from the signal and a decision is made by using a suitable classifier. The performance of this model relies on defining a set of features that pair up optimally to the designated classifier. Feature based classifiers have the advantage of being low in complexity as compared to decision theoretic based approaches. Since the objective was to develop a low complexity modulation classifier that also performs accurately in real time, the proposed work is based on the pattern-recognition approach.

The configuration of a generalized modulation classifier is shown in figure 3.1. The basic blocks can be implemented in a variety of ways depending upon the features and the modulation schemes to be identified.

Prior to the actual classification step, the preprocessing blocks performs translation to intermediate frequency, sampling etc. It also extracts a noise clean signal

to feed the feature extraction block to perform reliable modulation classification. The classifier section contains 2 main blocks; feature computation and classification. Feature computation is performed in two steps. First, selected features are extracted from the detected signal to get the parameters that will be used in discriminating different modulation types. Second, on the basis of such parameters, a decision is made about the modulation type in the classification block. Several techniques are available for this purpose, for e.g. linear classifiers, quadratic classifiers, tree-classifiers, neural network classifiers and Bayesian classifier etc.

3.2 Selecting Features

In the process of modulation classification the received signal contains much uncertainty which can be encountered by using statistical tools. A number of known statistics can be derived from continuous-time received signals which may be processed and used by the modulation classifier [31-53].

Figure 3.2 depicts some commonly used classification methods and features used in decision theoretic and pattern recognition based modulation classifiers.

The different types of decision theoretic classifiers were reviewed in chapter 2. Their approach and classes were also discussed. This section focuses on the pattern recognition approach and the common type of features used in the PR method. Following is the description of commonly used features for the Pattern Recognition approach.

1. Instantaneous property

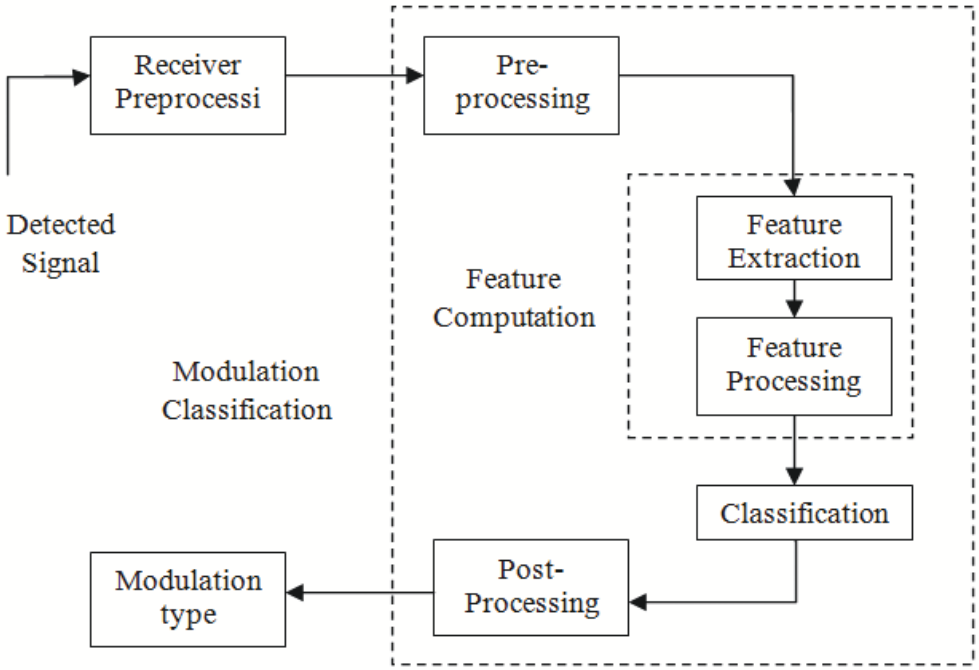

Figure 3.1: A general configuration of a modulation classifier

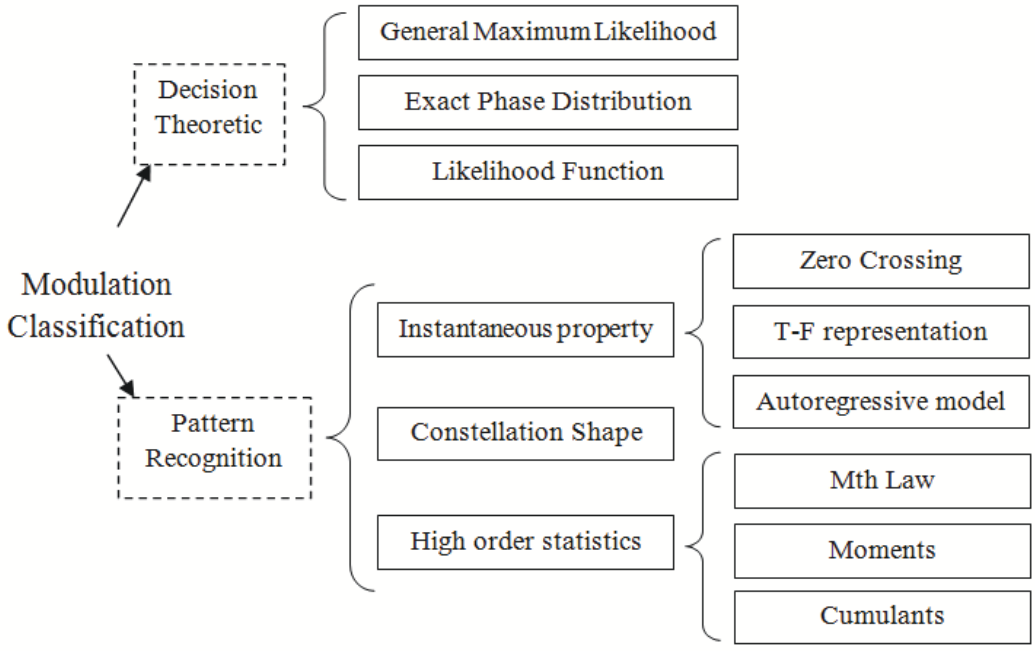

Figure 3.2: Commonly used classification methods and their respective feature sets

(a) Zero Crossing. This method calculates the zero-crossing instants of the signal, which evaluates the instantaneous frequency. This is commonly used to classify PSK and FSK modulations.

(b) Time-Frequency analysis. This method is based on the continuous wavelet transform. It performs quite well in classifying M-PSK, M-FSK and M-QAM modulation schemes.

(c) Autoregressive Model. This method is based on a particular autoregressive spectral estimation, used to classify PSK and FSK modulations.

2. Higher order statistics

(a) Moments. This method is based on analysis of the statistical moments of the detected signal. It performs well in identifying M-ASK and M-PSK modulations

(b) Mth Law. This method evaluates the analytical signal with the power of M. It is commonly used to identify M-PSK, MMSK and 2-FSK modulations.

(c) Cumulants. This method is based on the combination of the higher-order cyclic cumulants. It is effective in classifying M-ary QAM modulation schemes.

3. Constellation Shape This method is based on the calculation of a distance function between the symbols located in different parts of the constellation diagram. This method is capable of discerning 4, 16-QAM and 2, 4, 6-PSK

modulations.

The statistical moments can effectively characterize the joint PDF of the input signal feature [65]. This results in producing a more robust feature vector. Hence the higher order statistical moments were selected to be the features in the proposed modulation classifier.

The following section discusses moments in detail.

3.2.1 Statistical Moments

Higher order statistics are used to characterize the probability density functions and to estimate $f_X(x)$ from the signal under investigation. These statistics are determined by the characteristic function of X, given by

$$\Phi_X(\omega) \triangleq E\left[e^{j\omega X}\right] = \int_{-\infty}^{\infty} f_X(x) e^{j\omega x} dx \tag{3.1}$$

Which is the Fourier transform of $f_X(x)$ without the minus sign in the exponent. For the Gaussian random variable X with distribution $N(\mu, \sigma^2)$ the above characteristic function can be reduced to

$$\Phi_X(\omega) = exp\left(j\mu\omega - \frac{\sigma^2}{2}\omega^2\right) \tag{3.2}$$

Now, considering the definition of characteristic function from (3.1), the exponent function inside the expectation can be expanded to a power series

$$\Phi_X(\omega) = E\left[e^{j\omega X}\right] = E\left[\sum_{n=0}^{\infty} \frac{(j\omega X)^n}{n!}\right] = E\sum_{n=0}^{\infty} \frac{(j\omega)^n}{n!} m_n^X \tag{3.3}$$

$$= 1 + \frac{j\omega}{1!} m_1^X + \frac{(j\omega)^2}{2!} m_2^X + \frac{(j\omega)^3}{3!} m_3^X + \dots \tag{3.4}$$

Where m_n^X is the nth-order moment of X, i.e. $m_n^X = E[X^n]$. To find the nth-order moment of X we obtain the nth-order derivative of $\Phi_X(\omega)$ with respect to ω at point $\omega = 0$ as [3]

$$m_n^X = \frac{1}{j^n} \frac{^n}{(d\omega)^n} \phi_X(\omega) \ \ |_{\omega=0} \tag{3.5}$$

From this equation we can imply that the pdf $f_X(x)$ of a function can be characterized by the knowledge of the moments of X.

In higher orders, the central moments (moments about the mean) are more interesting than the moments about zero. The k_{th} central moment, of a real-valued random variable probability distribution X is

$$\mu = E\left((X - \mu)^k\right) \tag{3.6}$$

The first central moment is thus 0. The second central moment is also known as the variance, given by

$$Var(X) = \sigma^2 = E\left[(X - \mu)^2\right] \tag{3.7}$$

The third central moment, also known as the skewness, is the measure of the asymmetry or lopsidedness of a distribution [65]. The skewness is denoted by γ and defined as

$$Skewness = \gamma = E\left[X^3\right] - 3\mu\sigma^2 + 2\mu^3 \tag{3.8}$$

The fourth central moment, also known as the kurtosis, is the measure of the peakedness of a distribution [65]. The fourth central moment is always positive. The fourth central moment is denoted by κ and defined as

$$Kurtosis = \kappa = E\left[X^4\right] - 4\mu\gamma_1 + 6\mu^2\sigma^2 - 3\mu^4 \tag{3.9}$$

There exist moments beyond the 4th order that are also commonly used in AMI applications. However, these moments become increasingly difficult to calculate as they require more and more samples for correct estimation.

3.3 Selecting Classifier

As depicted in figure 3.1 the classifier block composed of 2 blocks. The first block, named feature extraction, was discussed in the previous section. The second block is called the classifier. The classifier is a function that discerns probabilities from training data in the process of classification.

There are a few comprehensive empirical studies in the literature, comparing learning algorithms [92]. Learning algorithms are now used in many practical domains,

and different performance metrics suit different domains. The performance metrics measure different tradeoffs in the predictions made by a classifier, and it is possible for learning methods to perform well on one metric, but be suboptimal on other metrics. A large scale empirical comparison of ten supervised learning algorithms using eight performance criteria were shown in [92]. They evaluated the performance of SVM, neural nets, logistic regression, nave bayes, memory-based learning, random forests, decision trees, bagged trees, boosted trees and boosted stumps. These classifiers were evaluated on eleven binary classification problems using a variety of performance metrics. These metrics are accuracy, F-score, lift, ROC area, average precision, precision/recall breakeven point, squared error, and cross entropy.

Among the many classifiers, the Bayesian classifier has provided a promising representation for modulation classification because it deals explicitly with issues of uncertainty and noise. Some of the most impressive results to date have come from this simple and much older classifier. Despite its simplistic approach, the Bayesian classifier has repeatedly proved to be competitive with more sophisticated algorithms [93-94].

Following is a more detailed review of the Bayesian classifier that is selected in the proposed work.

3.3.1 The Bayesian Classifier

It was noted from the literature that the Bayesian classifier provides a simple approach to represent probabilistic knowledge. The Bayesian method was designed for use in supervised learning [95]. The primary goal of the Bayesian classifier is to accurately predict the class of test instances in which the training data includes class information.

Let $\boldsymbol{C}$ be a random variable denoting the class of a test variable and let $\boldsymbol{X}$ be a vector of random variables denoting the extracted feature vector. Further, let $\boldsymbol{c}$ be a particular class label and $\boldsymbol{x}$ be a given test case to classify. One can simply use the Bayes rule to compute the probability of each class given the feature vector by

$$p(C = c|X = x) = \frac{p(C = c)p(C = c|X = x)}{p(X = x)} \tag{3.10}$$

Where $X = x$ represents the event that $X_1 = x_1 \Lambda X_2 = x_2 \Lambda X_k = x_k$. Since the probability $p(X = x)$ has no effect on the classification variable $\boldsymbol{C}$, it can be ignored. Eq. 3.10 becomes

$$p(C = c|X = x) \propto p(C = c)p(X = x|C = c) \tag{3.11}$$

Since the features are assumed to be conditionally independent, one obtains

$$p(C = c|X = x) \propto p(\Lambda_i X_i = x_i|C = c) \tag{3.12}$$

$$\propto \prod_i p(X_i = x_i|C = c) \tag{3.13}$$

which is very simple to compute for test cases. For continuous feature vectors we can write the conditional probability density function for a Gaussian distribution as

$$p(C = c|X = x) = \frac{1}{2\pi^{\frac{l}{2}}\sqrt{|R|}} exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T R^{-1}(\mathbf{x} - \mu)\right) \tag{3.14}$$

where $|R|$ is the determinant of an lxl covariance matrix. Covariance matrix R is

$$R = \frac{1}{K}\sum_{i=1}^{K}\left[\left(\mathbf{x}^i - \mu\right)^T\left(\mathbf{x}^i - \mu\right)\right] \tag{3.15}$$

where K is the number of patterns in the class and μ is the mean vector

$$\mu = \frac{1}{K}\sum_{i=1}^{K}\mathbf{X}^i \tag{3.16}$$

Taking 3.14 and converting the conditional pdf into log domain we get

$$log(p(X = x|C = c)) = -log(|R|) - (\mathbf{x} - \mu)R^{-1}(\mathbf{x} - \mu)^T \tag{3.17}$$

The constant terms in 3.17 were ignored since they would bear uniform effect on all the PDF.

Each class is characterized by the basic statistical parameters (mean vector, covariance matrix), which are computed from the training set. These parameters guide the discrimination process. Thus the above model leaves us with a small set of parameters to estimate from training data.

A common assumption in the Bayesian classifier is that values of continuous features are normally distributed [95]. Such a distribution is represented in terms of the mean and covariance matrix of the feature vector. In the proposed work, the Gaussianity of selected features was tested in chapter 4. All features, the first four moments, were tested and proven by several normality tests. Hence, it was deducted that the probability of a test value can be computed efficiently by using a Bayesian classifier and statistical moments.

3.4 The complete proposed model

The proposed classification framework is described symbolically in figure 3.3. The algorithmic part of the proposed modulation classifier framework was developed with professional modeling software. Whereas the final proposed algorithm for modulation classification was implemented and verified on Field Programmable Gate Arrays (FPGA). The following sections describe the framework in detail.

3.4.1 Signal Generation

The first stage in the framework is to generate the Amplitude, Phase and Frequency modulated schemes. Six different modulation schemes were generated, namely 2-ASK, 4-ASK, 2-PSK, 4-PSK, 2-FSK and 4-FSK, under different channel and noise conditions.

To produce a fair testing system, two sets of signals were created. First data set was used to train the system and the second data set for testing the system. Within these two categories, each modulation scheme was generated with 4 different Signal to Noise ratios. These SNR parameters were 2dB, 5dB, 10dB and 15dB SNR.

3.4.2 System Training

In this stage of the framework, the six modulation types were fed into the training system to first extract their features. These features will serve as the measurable quantities that make the classes distinct from each other. After the features were obtained they were arranged in the One-Vs-All (OVA) scheme. Details of OVA are explained later in the section. The OVA scheme was followed because the Bayesian classifier is a binary classifier and it can only distinguish between 2 classes at one instance of the experiment. Following is an illustration of the system training process

Let's consider the example of a single input waveform as shown in Figure 3.4; a noise corrupted 4-ASK modulation scheme. The training system samples this

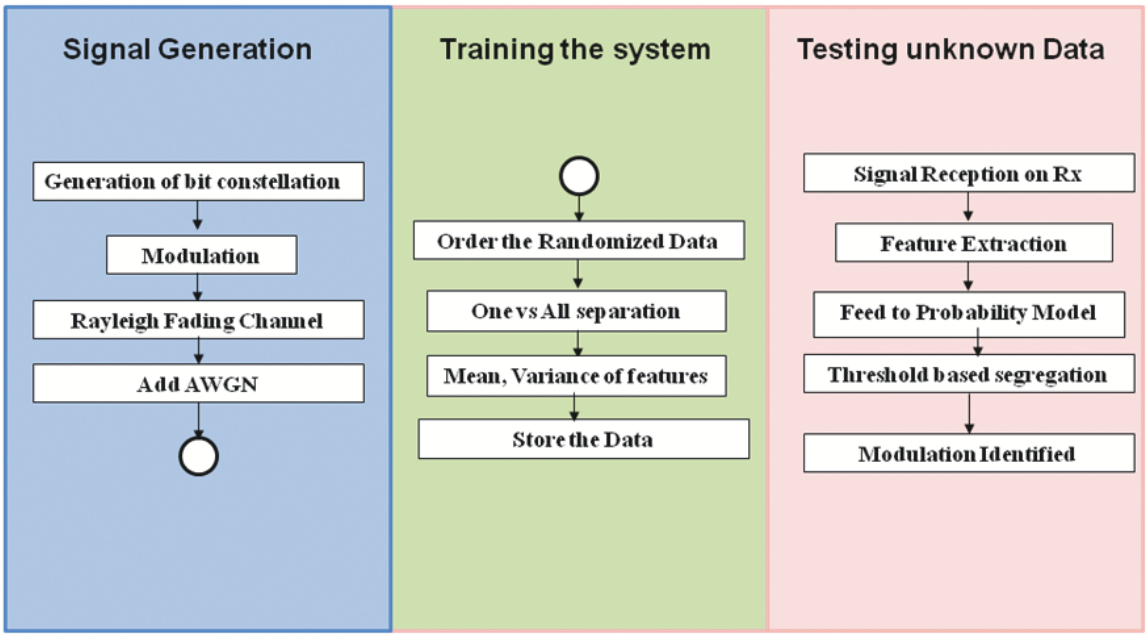

Figure 3.3: Overview of the proposed system framework

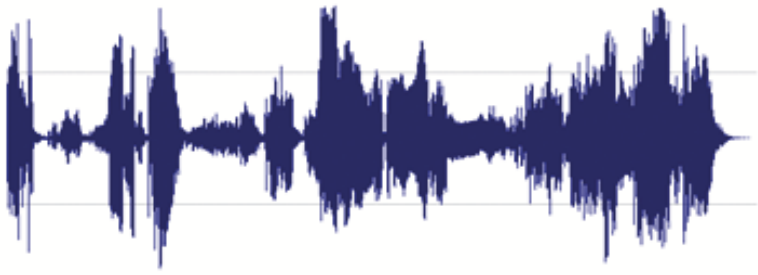

Figure 3.4: Generated training signal waveform

waveform and moves to the next step of feature extraction as shown below

In the feature extraction step, the first four moments, namely the Mean, Variance, Skewness and Kurtosis are calculated using the following expressions

1.

$$\mu_{signal} = \frac{1}{N}\sum_{i=1}^{N} X \quad whereN = sample\ \ size \tag{3.18}$$

2.

$$Var(X) = \sigma^2 = E\left[(X-\mu)^2\right] \tag{3.19}$$

$$= E\left[(X-\mu_{signal})(X-\mu_{signal)^T}\right] \tag{3.20}$$

$$= E\left[XX^T\right] - \mu^2_{signal} \tag{3.21}$$

3.

$$Skewness = \gamma = E\left[X^3\right] - 3\mu\sigma^2 + 2\mu^3 \tag{3.22}$$

4.

$$Kurtosis = \kappa = E\left[X^4\right] - 4\mu\gamma_1 + 6\mu^2\sigma^2 - 3\mu^4 \tag{3.23}$$

Next, the data is rearranged in an OVA setup. This is necessary because the Bayesian classifier only performs binary classification. Hence the feature data of ASK4 is arranged against the combined feature data of the remaining modulation schemes, as shown in figure 3.6. Such that in the testing stage, the unknown signal will be tested to see if it is close to ASK4 or the remaining ALL of modulation schemes. Since the Bayesian classifier works in binary, the result will be either

ASK4 or ALL. If the test resulted in the favor of ALL, it would be certain that the unknown signal is not ASK4.

Similarly, all the six modulation schemes will be provided to the training system. The system will create a data set of the extracted features and arrange them in OVA manner, as shown in figure 3.7.

This procedure of training the system by known signals is repeated several times. The ideal system training would have a large number of training runs with all the possible class variations. In the proposed classifier, the system is trained for 100 instances of each modulation scheme. Hence the trainer will accept the known signal, extract the features and arrange them in OVA, 100 times for each modulation scheme. As an illustration, the feature matrix of 100 training instances of ASK4 would look like the following

$$\underline{x}_{ASK4} = \begin{bmatrix} \mu_1 & \sigma_1^2 & \gamma_1 & \kappa_1 \\ & & \vdots & \\ \mu_{100} & \sigma_{100}^2 & \gamma_{100} & \kappa_{100} \end{bmatrix} \tag{3.24}$$

$$\underline{x}_{ASK4} = \begin{bmatrix} \mu_1 & \sigma_1^2 & \gamma_1 & \kappa_1 \\ & & \vdots & \\ \mu_{100} & \sigma_{100}^2 & \gamma_{100} & \kappa_{100} \end{bmatrix} \tag{3.25}$$

$$\underline{x}_{ASK4} = \begin{bmatrix} \mu_1 & \sigma_1^2 & \gamma_1 & \kappa_1 \\ & & \vdots & \\ \mu_{100} & \sigma_{100}^2 & \gamma_{100} & \kappa_{100} \end{bmatrix} \tag{3.26}$$

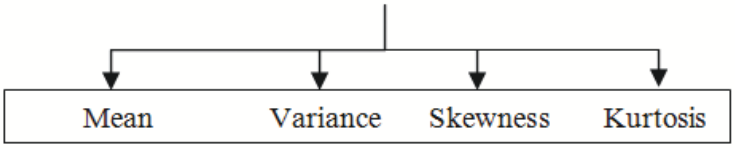

Figure 3.5: Extracting features from the training signal

Figure 3.6: OVA arrangement for the test case of ASK4 classification

Figure 3.7: OVA arrangement for all 6 modulation schemes

After all the features have been organized, the system calculates mean and covariance matrix of $\underline{x}_{ASK4}$ by the following expressions

$$\mu_{feature1} = \frac{1}{N}\sum_{i=1}^{N}\mu_{i,ASK4} \tag{3.27}$$

$$\mu_{feature2} = \frac{1}{N}\sum_{i=1}^{N}\sigma_{i,ASK4} \tag{3.28}$$

$$\mu_{feature3} = \frac{1}{N}\sum_{i=1}^{N}\gamma_{i,ASK4} \tag{3.29}$$

$$\mu_{feature4} = \frac{1}{N}\sum_{i=1}^{N}\kappa_{i,ASK4} \tag{3.30}$$

$$\mu_{feature4} = [\mu_{feature4}\ \ \mu_{feature4}\ \ \mu_{feature4}\ \ \mu_{feature4}] \tag{3.31}$$

$$Cov(\underline{x}_{ASK4}) = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_{i,ASK4} - \mu_{features})(\mathbf{x}_{i,ASK4} - \mu_{features})^T \tag{3.32}$$

where N = 100 instances.

3.4.3 Testing Unknown Data

At the testing stage, a different data set was used to keep complete anonymity. Another popular approach observed in literature was to divide the data set in two partitions of 70% and 30%. The 70% portion would be used for training the system and remaining 30% of data for the testing. There is very minor difference between the outcomes of the two methods. The division method is used if the data set cannot be custom generated. Following is a mathematical representation of the classification process performed on an unknown communication signal to

determine its modulation type.

Consider an unknown signal X as shown in figure 3.8, received and passed on to the modulation classifier for classification. The first step is to obtain the 4 features in order to compare them with the trained data set. Features will be extracted using the same procedure discussed in section 3.4.2.

Using the 4 features of the unknown signal and the previously trained feature parameters, the classification probability is calculated using 3.17. This probability will be the classification confidence value of the unknown signal being one of the six modulation schemes. The maximum probability from this vector shows the highest classification match to a certain class.

$$Log(P_i) = \sum_{i=1}^{6} -log(|Cov_i|) - (\underline{x}_i - \mu_{i,features})Cov_i^{-1}(\underline{x}_i - \mu_{i,features})^t \qquad (3.33)$$

where $i = \{1, 2, \ldots, 6\}\, \epsilon\, \{ASK2, PSK2, \ldots, FSK4\}$

3.5 Hardware Implementation framework

It was deduced that it is uncommon to find an AMC that performs well on low SNR and also can be realized into a compact module. A tradeoff exists between the classification accuracy and architecture complexity of the hardware implemented modulation classifier. Thus, a real-time modulation classification system

is highly sought after. Slight computational lag is usually tolerated depending on the application.

3.5.1 Selecting suitable platform

There are a few platforms available for the hardware implementation of mathematical algorithms. Some of the common platforms for the implementation of modulation classifiers are

1. General Purpose Processors (GPP)
2. Digital Signal Processing kits (DSP)
3. Field Programmable Gate Arrays (FPGA)
4. Rapid Radio Devices

A comparative study of the pros & cons of these platforms is illustrated in Table 3.1.

General Purpose Processors

General purpose processors (GPP) offer large amounts of program and user memory, floating point operation, better high level language development environments and are less expensive compared to dedicated DSPs. The rapid development in the processor technology has greatly improved the performance of some GPPs that outstrips the performance of dedicated DSPs.

According to a recent processor speed survey carried out by Berkeley design technology, a Pentium III processor outperforms the Texas Instrument TMS320C55xx, TMS320C62xx, TMS320C67xx, analog devices ADSP-21xx/219x, ADI-Intel MSA/Analog devices ADSP-2153x and the Motorola DSP563xx/DSP568xx [96]. But the general purpose processors are designed for general all purpose computing. Thus they must require certain specialized hardware attachments to complete the hardware testbed. This makes them more power demanding and less mobile than other specialized chips. Thus in the application of modulation classification, they can be only suited for ground based stations.

Digital Signal Processors

Digital signal processors offer a more attractive platform as compared to GPP since their instruction sets are optimized for performing repetitive or looping operations [73]. A DSP module consists of one or more signal processors and additional specialized hardware which enables the necessary DSP operations to be efficiently performed. Since in many communication systems, it is desired to have software updating capability, a DSP platform is ideal to use since device upgrading or updating can be easily achieved by downloading a new program into the processor memory. One of many applications where DSP is often used is data acquisition and processing in the real time.

However, DSPs have highly constrained memory capabilities for waveform storage and buffering. They also have highly complex assembly configurations that are required for fine hardware control.

Table 3.1: Common platforms for the implementation of modulation classifiers

	Reprogrammable	Real-time Performance	Ease to implement	Parallel Processing	Portability
RR	Yes	No	High	No	Low
GPP	Yes	No	Medium	No	Medium
DSP	Yes	Yes	Medium	No	Medium
FPGA	Yes	Yes	Low	Yes	High

Rapid Radio Devices

Rapid radio platform is designed to help those without any programming knowledge to rapidly implement their mathematical models on programmable hardware. Often the algorithm designer is unfamiliar with the process of hardware implementation. Certain hardware testbeds are developed to overcome this difficulty. The testbeds require a resource abundant hardware platform and Rapid deployment software [90]. This results in a time efficient hardware implementation of signal processing algorithms.

The problem with Rapid Radio is that the translations are almost impossible to interpret. Hence any low level programming and fine control is off limits. Other drawbacks include low processing speeds, lack of accuracy and dependence on vendor software [91].

In the case of modulation classification, if development speed is paramount, then an operational but possibly suboptimal classifier can be built in a short time. It may be derived from the above comparison that DSPs offer the right set of solutions for implementation when compared to GPP and Rapid radios. But there is another platform called Field programmable gate array (FPGA) that has an

edge over DSPs in modular design flow. FPGAs can implement logic with parallel functioning modules while keeping a low power consumption profile. The programmer can design very high speed interconnection paths between modules using low level programming. This results in very high performance specifications [97]. The development does require some programming background and basics of computer architecture. But FPGA provides the right performance and control that we require for the proposed modulation classification algorithm. Hence FGPAs are used in the hardware implementation of the proposed algorithm.

3.5.2 Objectives of the Implementation

As the framework for classification of digitally modulated signals has been developed, the next stage of the thesis is to implement the developed algorithm on testbed hardware. The proposed classifier is to be implemented and evaluated for real time performance. The choice of platform was deducted to be FPGAs. By using hardware descriptive language, a multi-core and efficient architecture is to be designed and implemented. The algorithm should be tested in real time and all the functions of the proposed modulation classifier should be verified on the FPGA.

3.5.3 Integral components of an FPGA

FPGA is a reconfigurable or reprogrammable hardware platform to develop and run programs and applications. FPGA can be best described as a sea of logic gates

that are interconnected to each other in order to produce any logical calculation. Alongside with logic combinations FPGA can be configured to run sequential or algorithmic procedures. An FGPA is often employed by designers to verify the functionality of their programs and algorithms [91]. This practice provides an insight into the possibility of practical realization of the design.

Certain components of the FPGA worth mentioning are as follows

- **Combinational Logic Blocks (CLB) & Look-up-Table (LUT)** FPGAs are basically composed of Configurable Logic Blocks (CLB), Interconnecting wires and I/O pads. As depicted in figure 3.9 these components make up the functional resources that a FPGA can offer. FPGA vendors provide sufficient interconnections between the CLBs and large enough LUTs to encompass a large variety of designs.

- **Common Clock and Maximum Frequency** A clock in any digital circuit is defined as the heartbeat of the system. In FPGAs this is no different. Although the clock identifies the maximum system frequency but the actual highest achievable operational speed is determined by the user combinational blocks [85]. Hence the overall speed is as high as the slowest module/block in the user architecture.

  The onboard oscillator generates the system clock. Assuming the clock speed is 50MHz frequency. That translates to time duration of 20ns per clock cycle. This means that an operation taking 4 clock cycles to complete will then take duration of 80ns.

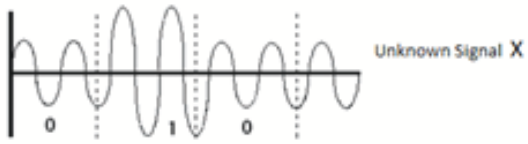

Figure 3.8: Unknown signal waveform

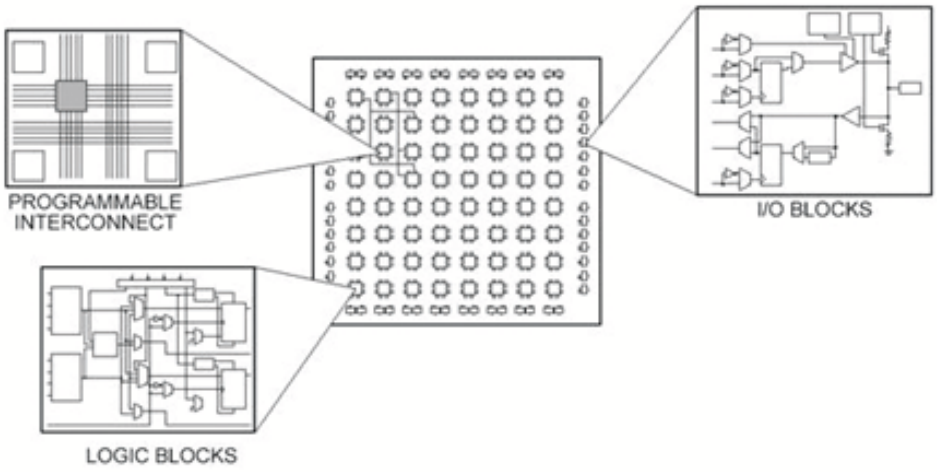

Figure 3.9: Layout of Configurable Logic Blocks inside an FPGA

3.5.4 Area, power and speed tradeoffs

In designing logic blocks or logic solving modules the design suite makes use of the CLBs. Bigger the design, greater the number of Logic blocks will be used. Also, as the numbers of active logic blocks are increased the power requirement also increases [85]. Hence there exist several tradeoffs in designing and implementing large architectures. This power-to-functionality tradeoff is an essential design constraint in portable computer design. If the algorithm to be implemented is targeted for portable media then power consumption as well as resource utilization becomes a major factor.

In computer simulation or mathematical modeling, the computational speed is often ignored and countered by faster computers. But this becomes a challenge when FGPAs are in question. FPGAs have limited resources and they are targeted to run on limited power sources. An ideal architecture should run the algorithm with minimum number of clock cycles, minimum power requirement and taking minimum area possible.

Most of the algorithms are sequential or recursive in nature and require the completion of certain parameters before moving forward. This generates system bottlenecks and speed constraints on the performance. This problem is solved by the method of parallel processing, as explained in the next topic.

3.5.5 Parallel Processing

Every logic module in an FPGA is created by configuring the CLBs. There can be more than one data path on which a single part of the algorithm is performed. To relieve the congestion in an algorithm or to utilize the available area efficiently, non recursive computations can be divided into parallel modules, as shown in figure 3.10. This practice makes use of the available CLBs and divides the work load into several parts. Figure below illustrates this process by showing several input ports and data streams connected to several different modules. The processes are divided into standalone computations and the results transferred to the next stage of the algorithm.

3.5.6 Necessity of Floating point operations

The representation of integers and decimal numbers in FPGA is a very difficult task by itself [98]. To resolve this issue a microprocessor standard called the IEEE-754 is followed that maps floating point numbers into binary digits. The accuracy of the conversion can either be single precision or double precision. The bit-widths of single and double precision numbers are 32 bit and 64 bits respectively.

IEEE754

The IEEE 754 is a 32 bit representation of a floating point numbers. Floating point numbers are used to represent real values in digital computations. The IEEE 754 standard is one way of representing floating point numbers, among many [98]. Floating point numbers contain an exponent and a mantissa such that

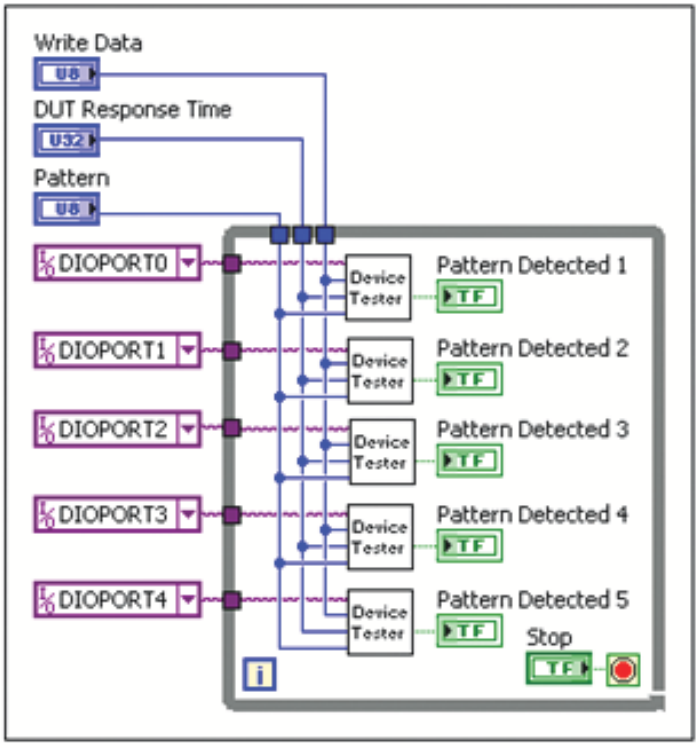

Figure 3.10: Example of parallel processing using multiple modules at same time

Floating point number (n) = radix exponent x mantissa

The IEEE 754 standard defines two degrees of precision

1. Single precision (32 bits)

2. Double precision (64 bits)

We will use the single precision format in our design, as depicted in figure 3.11, because it sufficiently addresses the accuracy requirements and a double precision approach will relay unnecessary overhead.

The single precision format is designed as follows

$$FloatingPointValue = (-1)^S 2^e \times 1.f(normalized) where E > 0 \ else \quad (3.34)$$

$$= (-1)^S 2^{-126} \times 0.f(denormalized) \quad (3.35)$$

Where

$f = (b_{23}^{-1} + b_{22}^{-2} + b_i^n + \cdots + b_0^{-23})\ where\ b_i^n = 1\ or\ 0$

$S = sign(0\ is\ positive;\ 1\ is\ negative)$

$E = biased\ exponent; E_{max} = 255, E_{min} = 0, E = 255\ and\ E = 0\ for\ special\ values$

$e = unbiased\ exponent; e = E - 127(bias)$

In summary, it was deducted that the Bayesian classifier is a simple and efficient approach to the problem of automatic modulation classification. However, it typically relies on an assumption that continuous feature vectors follow a Gaussian distribution, which may not hold for some domains. But our choice of features

hold Gaussianity and this is proven in chapter 4. This combination presents a low complexity and accurate system model for automatic modulation classifier.

Also investigated in this chapter were possible candidates for the hardware implementation of the proposed AMI algorithm. It was concluded that the field programmable gate arrays (FPGA) platform is an ideal candidate for a real-time implementation of this algorithm. The implementation details are presented in chapter 5.

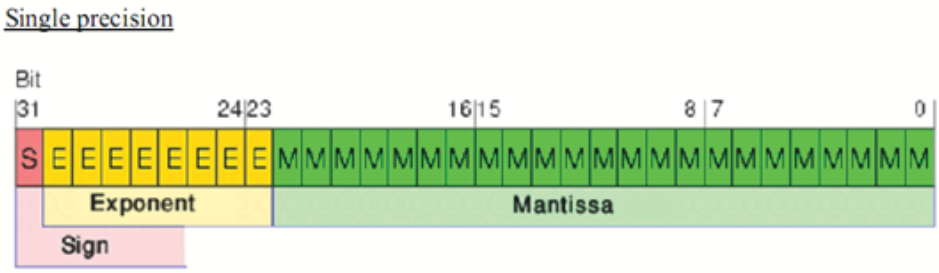

Figure 3.11: Frame of an IEEE754 floating point unit data value

CHAPTER 4

TESTING AND EVALUATION OF THE PROPOSED SYSTEM

In this chapter the performance parameters and classification accuracy of the proposed modulation classifier are discussed. This section contains the simulated performances and algorithm results. Certain thresholding methods are also discussed that help to improve the classification accuracy.

The main goal for the experiment was to test and asses the proposed classification methodology under controlled, yet realistic conditions. An ideal modulation classifier should be highly accurate in predicting the correct modulation types and it should do so by observing small quantities of sampled data. The classifier should be computationally inexpensive, its performance should not degrade under harsh channel conditions, the range of identifiable modulations should be large and the method should be implementable in real time. Also the classifier should perform the above mentioned duties consistently on low signal to noise ratios.

The digital modulation types produced for the simulations were 2PSK, 4PSK, 2FSK, 4FSK, 2ASK, 4ASK. The training and testing of the proposed classifier is based on randomized signal sequences. This is because the classification algorithms must not be sensitive to the content itself. This also ensures that each data stream used for modulation of the radio signal was unique.

The symbol rate was chosen to be $f_d = 2500Hz$ because it should be an integer factor of the sampling rate. The phase states used in the generation of the 2PSK signals were $\phi = 0$, $\phi = \pi$. For the 4PSK signals, the phase states were $\phi = 0$,$\phi = \frac{\pi}{2}$,$\phi = \pi$, and $\phi = -\frac{\pi}{2}$. The frequency deviation for the 2fSK signals was $f_\Delta = 2500Hz$, and $f_\Delta = 1250Hz$ for the 4FSK signals.

4.1 Classification accuracy

Measuring the accuracy of a modulation classifier is not a straight forward task. There are a number of subtle reasons for this phenomenon. Initially all modulation classifiers are described on the measure of how many signals they correctly classify. But a practical modulation classifier would have to also be low in complexity. Most of the modulation classifiers are designed to handle specific unknown parameters. These parameters distinguish the classifier to be blind or semi-blind. Then there are different classes to be identified in different modulation classifiers. This makes it difficult to follow a benchmark performance to show the effective worth of a classifier. Also, performance of different classifiers also cannot be compared, unless the candidate modulations are the same. Secondly, one cannot really com-

pare their performances unless the uncertainties of the modulation classifiers are the same. One has to consider the above mentioned parameters when comparing the accuracy of any modulation classifier.

This section shows only the simulated accuracy of the classifier under different conditions. The classifier accuracy was also improved by applying post classification thresholding. Later sections show how the proposed classifier is low in complexity and a good candidate for hardware implementation.

Table 4.1 depicts performance of the proposed modulation classifier at 10dB SNR. It was seen that the classifier performs very well in four of the six classes. The highest uncertainty was for the case of Frequency Shift Keying (FSK) modulation. The proposed modulation classifier was unable to distinguish between FSK2 and FSK4 due to close proximity in their features. Small uncertainties were also present between PSK2 and PSK4 modulation schemes.

The problem of misclassification in Frequency Shift Keying modulation has been alleviated by using post classification thresholds. Thresholds are similar to the features extracted from the signals but they are never included in the feature set. The basic idea of using a threshold is that they help in separating very few classes from each other. For this reason they are not very useful for all the classes and thus used in the post classification scenarios.

Following section explains the parameters selected for thresholding and their results.

4.2 Handling uncertainties in classification by thresholding

1. **Zero Crossing**

   Zero crossing is a count of how many times any given signal crosses the point of zero amplitude. Zero crossing is a very commonly used measure in electronics and image processing. In image processing this method is used to detect sharp edges in an image. This is done to detect the boundary of an object or to identify the discontinuities in an image. In terms of hardware realization this can be thought of as the point where the positive swing of the alternating signal ends and the negative swing starts.

   In the proposed algorithm of modulation identification we applied the zero crossing counters to distinguish between FSK2 and FSK4. This was needed because the features of FSK2 and FSK4 were very close and very difficult to identify. Thus, the zero crossing method proved to be the distinguishing factor in sub-FSK classification. When applied to the input signal, the ZC count for FSK4 was always greater than that of FSK2. This is because FSK4 operates on 4 frequencies of which 2 frequencies are at higher positions in the spectrum.

2. **Akaike Information Criterion** The Akaike Information Criterion (AIC) is a goodness of fit measure of a statistical model [14].The AIC is used in quantifying and comparing different models. Different statistical models

may have different AIC. The criterion is based on selecting the order that minimizes the error

$$AIC(p) = \ln \widehat{\sigma}_{wp}^2 + \frac{2p}{N} \tag{4.1}$$

where $\widehat{\sigma}_{wp}^2$ is the variance between the Auto regressions of a statistical data set at two different values of sample spacing. Autoregressive spectrum estimation is an alternative to Fourier analysis for obtaining the frequency spectrum of a signal [60]. The method used here for autoregressive spectrum analysis consists of estimating the auto correlating coefficient followed by an inversion of the auto correlation matrix. Given an input signal

$$x(k) = s(k) + n(k) \tag{4.2}$$

Autoregressive spectrum modeling can be accomplished by solving the following system of equations

$$\begin{bmatrix} R_{xx}(0) & R_{xx}(1) & \dots & R_{xx}(N-1) \\ R_{xx}(1) & R_{xx}(0) & \dots & R_{xx}(N-2) \\ \vdots & \vdots & \vdots & \vdots \\ R_{xx}(N-1) & R_{xx}(N-2) & \dots & R_{xx}(0) \end{bmatrix} \tag{4.3}$$

$$\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_4 \end{bmatrix} = \begin{bmatrix} R_{xx}(1) \\ R_{xx}(2) \\ \vdots \\ R_{xx}(N) \end{bmatrix} \tag{4.4}$$

Where the auto correlation estimates $R_{xx}(k)$ are found as

$$R_{xx}(k) = \sum_{n=0}^{M} x(n)x(n+k) \tag{4.5}$$

Where M in (4.5) represents the number of samples in the analysis frame. In (4.4) the a vector represents the coefficients for the polynomial that best fits the frequency spectrum.

We have used the AIC to distinguish between PSK2 and PSK4 modulation schemes. The PSK 2 scheme has lesser variance in its auto regressive model as compared to the PSK-4 scheme. Thus a threshold τ_1 can be used to differentiate between the two modulation schemes.

Table 4.2 shows the modulation classification results at 10dB SNR after applying the two thresholds. It can be seen that after thresholding the FSK 2 and FSK4 modulation schemes are clearly set apart by the zero crossing parameter. The confusion between PSK2 and PSK4 has also been removed to a certain degree.

Table 4.3 depicts the contingent table for a bigger range of SNR. The results shown in Table 4.3 are post thresholding. It can be observed that even at the lowest SNR of 2dB the modulation classifier performs quite well. In summary,

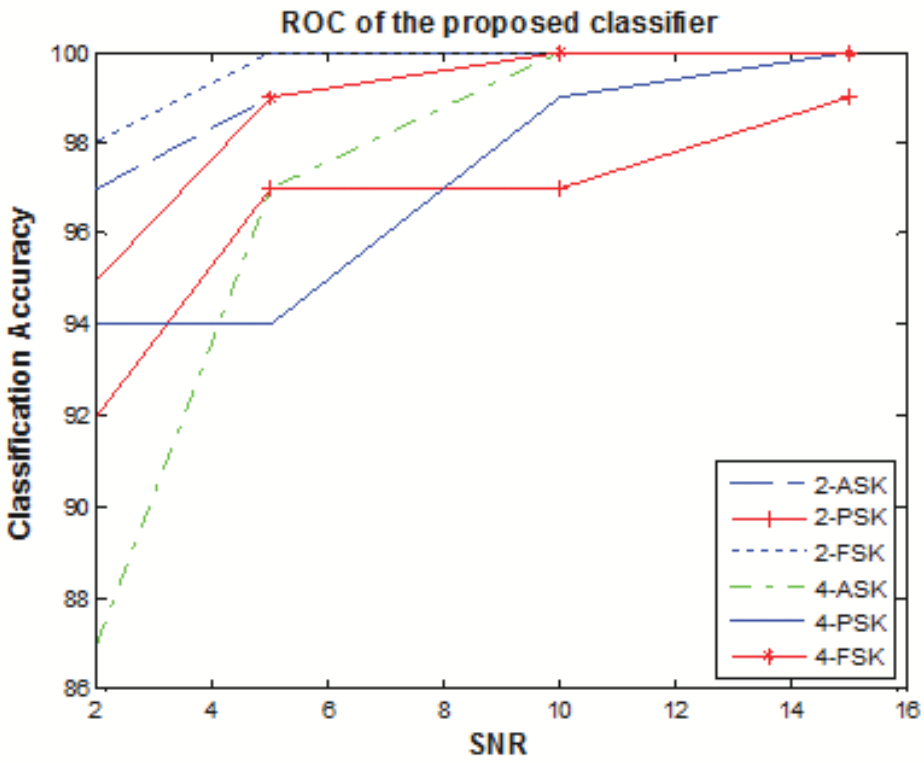

Figure 4.1: ROC of the proposed modulation classifier

after calculating the two thresholds and evaluating the accuracy the overall performance of the modulation classifier was increased.

Figure 4.1 illustrates Low SNR classification results in the form of ROC. The ROC gives detailed behavior of the proposed modulation classifier.

4.3 Gaussianity Test

The probability model of the Bayesian classifier used in the proposed AMI is Gaussian. Previously it was mentioned that the probability model assumes the values of continuous features as normally distributed. This means that the 4 features used in the classifier should satisfy the normality tests. There are many

ways to test a distribution for Gaussianity including histogram plots, hypothesis testing using cumulants and other goodness of fit tests. Hence, each 100 instance feature vector is tested for Gaussianity and the results are as follows.

Figure 4.2, 4.3 and 4.4 show the histogram plot of the 4 features of ASK, PSK and FSK respectively. It can be observed from the histogram plots of the four features that they follow Gaussian distribution with certain degree of freedom.

To further prove the Gaussianity we evaluated two of the most popular Gaussianity tests, the Jarque Bera test and Lillie fors test.

4.3.1 Jarque bera test

The Jarque-Bera test is a test of normality that evaluates a sample distribution as Gaussian or not. This test is based on the sample kurtosis and skewness with the assumption of unknown mean and variance [99]. The test is defined as

$$JB = \frac{n}{6}\left(S^2 + \frac{(k-3)^3}{4}\right) \tag{4.6}$$

Where n is the number of observations (or degree of freedom), S is the skewness and K is the kurtosis. They are defined as

$$S = \frac{\mu_3}{\sigma_3} = \frac{\mu_3}{(\sigma^2)^{\frac{3}{2}}} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^{\frac{3}{2}}} \tag{4.7}$$

$$K = \frac{\mu_4}{\sigma_4} = \frac{\mu_4}{(\sigma^2)^{\frac{3}{2}}} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2} \tag{4.8}$$

	2ASK	2PSK	2FSK	4ASK	4PSK	4FSK
2ASK	100	0	0	0	0	0
2PSK	0	97	0	0	3	0
2FSK	0	0	50	0	0	50
4ASK	0	0	0	100	0	0
4PSK	0	1	0	0	99	0
4FSK	0	0	50	0	0	50

Table 4.1: Contingent Classification without Thresholding at SNR 10dB

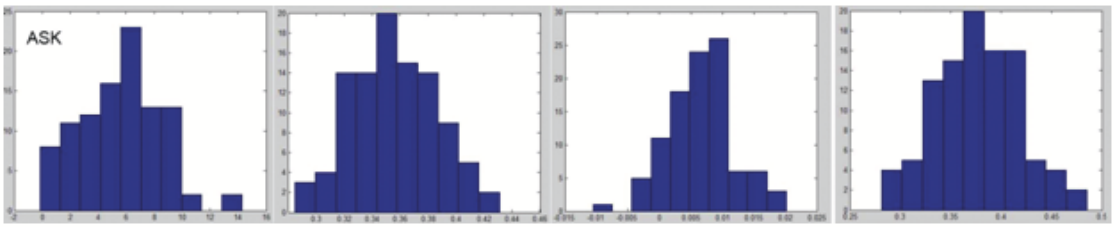

Figure 4.2: Features extracted from ASK modulation

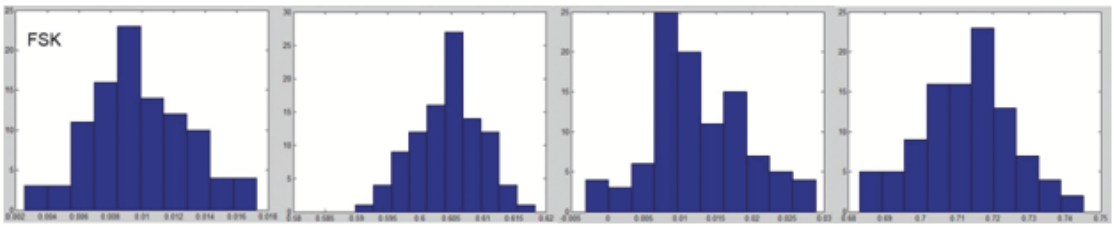

Figure 4.3: features extracted from FSK modulation

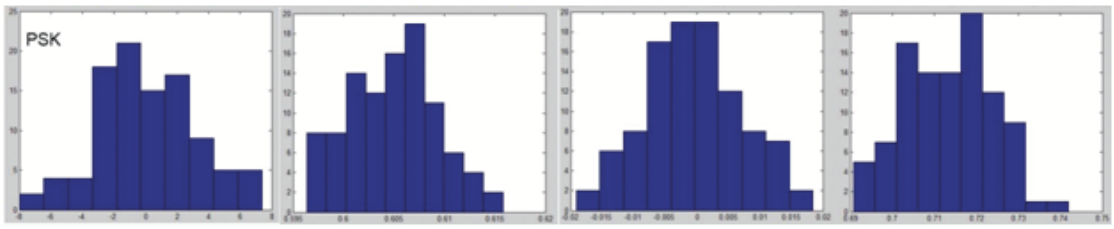

Figure 4.4: features extracted from PSK modulation

	2ASK	2PSK	2FSK	4ASK	4PSK	4FSK
2ASK	100	0	0	0	0	0
2PSK	0	99	0	0	1	0
2FSK	0	0	100	0	0	0
4ASK	0	0	0	100	0	0
4PSK	0	1	0	0	99	0
4FSK	0	0	0	0	0	100

Table 4.2: Contingent table at 10dB SNR after applying zero crossing and AIC thresholds

	SNR=2dB	SNR=5dB	SNR=10dB	SNR=15dB
2ASK	97	99	100	100
4ASK	87	97	100	100
2PSK	92	97	98	99
4PSK	94	94	99	100
2FSK	98	100	100	100
4FSK	95	99	100	100

Table 4.3: Performance (%) of the classifier using ZC and AIC thresholds

Where μ_3 and μ_4 are the third and fourth central moments.

4.3.2 Lillie fors test

The Lilliefors is also a goodness-of-fit test used to evaluate the null hypothesis of data coming from a normally distributed population [99]. The Lillie fors test was also passed for all features in all modulation schemes.

4.4 Robustness of the Selected Features

In this section we will investigate the robustness of statistical moments when used as features for modulation classification. This is done by using classifiers other than the one proposed in this work and use statistical moments as their features. The Neural networks and the Nearest Neighbor classifier were tested and their

	2ASK	4ASK	2PSK	4PSK	2FSK	4FSK	C%	E%
2ASK	100	0	0	0	0	0	100	0
4ASK	0	96	0	4	0	0	96	4
2PSK	0	0	100	0	0	0	100	0
4PSK	0	0	22	78	0	0	78	22
2FSK	0	0	0	0	66	34	66	34
4FSK	0	0	0	0	84	16	16	84

Table 4.4: Contingent table - results of ANN classification with SNR=5dB

respective results are shown in the following.

4.4.1 Neural Networks Classifier

Artificial Neural Networks (ANN) is an information processing system or algorithm that is inspired to work in the same way as the brain processes information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Neural networks can be applied in a large number of applications in which a functioning relationship exists between the input and the output. Neural networks are seen to work quite well even if these relationships are highly complex and difficult to functionalize [92].

By analyzing the modulation classification results tabulated in Table 4.4, it was seen that the selected features perform well in the ANN classifier.

Correctly Classified Instances 456 (76%)

Incorrectly Classified Instances 144 (24%)

	2ASK	4ASK	2PSK	4PSK	2FSK	4FSK	C%	E%
2ASK	100	0	0	0	0	0	100	0
4ASK	0	97	0	3	0	0	97	0
2PSK	0	0	96.3	3.7	0	0	96.3	3.7
4PSK	0	0	5.7	94.3	0	0	94.3	5.7
2FSK	0	0	0	0	37.5	62.5	37.5	62.5
4FSK	0	0	0	0	47.8	52.2	52.2	47.8

Table 4.5: Contingent table - results of k-NN classification with SNR=5dB

4.4.2 Nearest Neighbor Algorithm (NN)

$$JB = \frac{n}{6}\left(S^2 + \frac{(k-3)^3}{4}\right) \tag{4.9}$$

Similarly the nearest neighbor classifier was also computed using the statistical moments as features. Since the NN classifier is derived using Gaussian probability, the NN classifier will perform similar to the Bayesian classifier. The nearest neighbor classification algorithm is the simplest of all machine learning algorithms. A test case subject is classified by the maximum number of previously classified neighbors. The test subject is then classified as the class that is most common within its neighbors [92]. If k = 1, then the object is simply assigned to the class of its nearest neighbor.

The classification Results agree to the postulated theory as illustrated in Table 5

Correctly Classified Instances 477 (80%)

Incorrectly Classified Instances 123 (20%)

4.5 Different Effects degrading the System Performance

As a signal propagates from the transmitter to a receiver in any given environment the signal waveform faces certain damaging effects. We are always surrounded by objects may it be urban or natural. It is well known that the electromagnetic waves can travel through any non metal objects but the degree of similarity between the incident and the reflected/refracted waveform depends on the relative permittivity ε_r of the obstacle. Certain other natural phenomena like rain, humidity, snow, heavy smoke or volcanic dust might also degrade the signal waveform [100].

When such conditions are faced by the communication signal, it can be bent, absorbed or attenuated to a degree depending on the frequency, angle, material composition etc.

4.5.1 Channel Modeling

Channel is the route that a transmitting waveform takes to reach the receiver. This route can be thought of as a path in 3d vector space of our environment on which the transmitted waveform travels. Channel can be any transmitting medium that can convey a waveform from the transmitter to the receiver. It can be a cable, fiber optic line or a wireless medium. Every channel, regardless of its nature also exerts certain effects on the signal waveform. The degree of the effects can be specific to the type of channel in question.

Our study will include the behavior of a wireless channel. A wireless channel unlike

other type of channels can change its behavior on the course of its propagation. For example we can set off in a rural environment and contact a person who is in an urban environment. There may be many substations in the route connecting to each other to maintain communication. Nevertheless the environment of the two locations cannot be correlated without any real time data feed. Both routes will be random vector spaces and the channel effects will then also be random variables. In the case of line of sight antennas certain characteristics of the channel will depend on the nearby functions such as buildings and the antenna location. Hence depending on the path taken by the signal, the wireless channel effects may change.

There are several attempts to model this behavior of the channel [100]. These models provide the channel behavior in the form of an impulse response. To obtain the intensity of the received signal that has suffered channel effects we intend to capture the power profile of the received signal. In order to do so we can convolute the channel impulse response with the transmitting signal power profile to obtain the PDP of the received signal.

There are 3 major effects that a channel can induce on a traveling signal that are

1. **Shadowing** Shadow fading or shadowing is the phenomena when there exists a signal blocking obstacle between the signal and the receiver. This results in a major loss of signal strength at the receiver side.

2. **Path Loss** Path loss is a natural degradation of electromagnetic wave signal strength due to the inverse square law. The law postulates that the signal

strength is inversely proportional to the square of the distance it has traveled [101]. Path loss is not limited to only free space attenuation it can also be due to refraction and absorption in the given path.

3. **Mutipath** This phenomenon is specific to the wireless communication channels. Since a signal traveling in free space will eventually hit any obstacle that resides in its path. This collision is imminent and cannot be avoided due the dispersive radiation pattern of commonly used transmitting antenna and the increasing number of urban areas in the world. As shown in figure 4.5 when the collision happens the signal will bounce off the obstacle because the metal quantity in urban buildings is quite high. This results in a scenario where multiple instances of the same transmitted signal will propagate in free space and will reach the receiver from different angles.

4.5.2 Fading effects

During propagation the communication signal may be scattered between the transmitter and receiver. In this scenario there is no single route that dominates communication interchange. The most obvious path is the direct line of sight (LOS) path. However there will be many objects surrounding the direct path. These objects may serve to reflect or refract the signal. As a result, there may be many non-LOS paths through which the signal may effectively travel to the receiver [101].

When the transmitted signal arrives at the receiver, the effective radio wave

is a summation of all the reflected/refracted signals. These signal waveforms traveled by multiple paths through the channel. The multipath signals will all sum together at the receiver resulting in an attenuated version of the original signal. The attenuation is caused by the difference in the signal phases induced by the obstacles in the channel. The change in signal phase is a function of path length and relative permittivity of the obstacles. Slightest difference in phase can mitigate the signal amplitude to a significant level. Often the resultant waveform is attenuated version of the original signal.

Flat Fading vs. Freq Selective Fading

Flat fading is said to be experienced when all the frequency components of the signal are subjected to the same amount of fading. Whereas in frequency selective fading different frequency components will go through different fading effects that may or may not correlated.

Slow fading vs. fast fading

Slow fading occurs when the change in signal amplitude and phase is almost constant over a period of time. This scenario can be related to a stationary transmitting and receiving device with a constant environmental setup. Whereas in fast fading the amplitude and phase of the signal changes greatly over a period of time. This fading phenomenon can be correlated to traveling transceivers in a dense urban environment. The signal thus experiences a great deal of change in the fading effects over a period time.

Rayleigh Fading

The Rayleigh fading model is particularly useful in scenarios where the transmitter or receiver is often in motion. This behavior of the transmitter/receiver causes the signal path lengths to vary, resulting in a change of the signal level [102]. This occurs due to the continuously varying path lengths of the communication signal. The Rayleigh fading model can be used to analyze radio signal propagation on a statistical basis. It operates best under conditions when there is no direct line of sight signal. Many cellular telephonic communications in dense urban environment fall into this category. Other examples of non line-of-sight communication include ionospheric propagation. In this scenario the communication signal reaches the receiver by very large number of individual routes. Signal propagation by tropospheric ducting also depicts the same behavior. All of these examples are ideal for the use of Rayleigh fading channel model.

Single Tap

The term tap refers to the number of multipath reflections that a signal has experienced. Single tap means that second to an original source, there will be another multipath reflected signal that arrives at the receiver after a certain time delay ?. Single tap data can provide quite some insight to observe the channel behavior. Higher level taps reach the receiver with such low amplitudes that their impact becomes insignificant.

Multi Tap

Multiple taps in a fading environment can provide a deeper more accurate measure of the channel behavior. This can be used in developing more robust

Channel	Terrain Type	Doppler Spread	Spread	LOS
SUI-1	C	Low	Low	High
SUI-2	C	Low	Low	High
SUI-3	B	High	Low	Low
SUI-4	B	High	Moderate	Low
SUI-5	A	Low	High	Low
SUI-6	A	High	High	Low

Table 4.6: Terrain Type and Doppler Spread for SUI Channel Models

algorithms and choosing more effective features from the signal. According to the Stanford University Interim (SUI) channel models the behavior of 3 taps are enough to enclose the behavior of channel effects [100]. The different terrain types and channel parameters are depicted in table 4.6.

Table 4.7 and table 4.8 show the SUI-2 and SUI-3 channel model parameters respectively. These parameters are evaluated on the basis of different urban environments. Parameters such as power delay profile, Doppler shift, antenna types etc are shown based on the height and spacing of the buildings.

Table 4.7: SUI - 2 Channel Model

	Tap 1	Tap 2	Tap 3	Units
Delay	0	0.4	1.1	μs
Power (Omni ant.)	0	-12	-15	dB
90% K-factor (omni)	2	0	0	dB
75% K-factor (omni)	11	0	0	dB
Power (30^o ant.	0	-18	-27	dB
90% K-factor (30^o)	8	0	0	dB
75% K-factor (30^o)	36	0	0	dB
Doppler	0.2	0.15	0.25	Hz
Antenna Correlation	0.5	0.5	0.5	
Gain Reduction Factor	2	2	2	dB
Normalization Factor	-0.3930	-0.3930	-0.3930	dB

As in the case of cellular telecommunications, the mobile station and the base

station are both effectively composed of omni directional antennas, hence both these SUI models cater for the parameters related to omni directional antennas. Most applications of automatic modulation classification are composed of omni directional transmit and receive antennas.

Table 4.8: SUI - 3 Channel Model

	Tap 1	Tap 2	Tap 3	Units
Delay	0	0.4	0.9	μs
Power (Omni ant.)	0	-5	-10	dB
90% K-factor (omni)	1	0	0	dB
75% K-factor (omni)	7	0	0	dB
Power (30^o ant.	0	-11	-22	dB
90% K-factor (30^o)	3	0	0	dB
75% K-factor (30^o)	19	0	0	dB
Doppler	0.4	0.3	0.5	Hz
Antenna Correlation	0.4	0.4	0.4	
Gain Reduction Factor	3	3	3	dB
Normalization Factor	-1.511	-0.511	-0.511	dB

Classification results in multi-tap Rayleigh fading channel

In our case we have selected the SUI-2 and 3 models to observe the robustness of our modulation classifier in Rayleigh fading channel. The goal is to first observe the channel effects then improve the modulation classifier to sustain the accuracy in damaging channel conditions. The effects of Rayleigh fading channel were applied to the unknown signal waveform and then fed to the AMI algorithm. Table 4.9 shows the classification accuracy for single tap Rayleigh fading channel in 10dB SNR conditions. It can be seen that without thresholding, the fading channel has reduced the classification accuracy considerably.

Table 4.10 shows the classification accuracy of the proposed modulation classifier under 2-tap Rayleigh fading channel in 10dB SNR conditions. The highest

1-tap	2ASK	2PSK	2FSK	4ASK	4PSK	4FSK
2ASK	81	0	0	19	0	0
2PSK	0	52	28	0	0	20
2FSK	0	50	25	0	0	25
4ASK	31	0	0	69	0	0
4PSK	0	0	0	0	100	0
4FSK	0	17	27	0	0	56

Table 4.9: Contingent table of classification under 1-tap Rayleigh fading channel

2-taps	2ASK	2PSK	2FSK	4ASK	4PSK	4FSK
2ASK	76	0	0	24	0	0
2PSK	0	43	25	0	0	32
2FSK	0	51	24	0	0	25
4ASK	20	0	0	80	0	0
4PSK	0	0	0	0	100	0
4FSK	0	18	35	0	0	47

Table 4.10: Contingent table of classification under 2-tap Rayleigh fading channel

numbers of taps investigated in this work are 2-taps. Increasing the number of taps any more will have minimal effect on the signal as the 3rd tap signal is severely mitigated [100].

Classification with Channel effects and thresholding Table 4.11 shows the contingent table for classification accuracy of the proposed modulation classifier including the Zero Crossing and Akaike Information Criterion thresholds.

It is seen that the classification accuracy has been considerably increased. Amplitude shift keying modulation scheme has been the most affected of the six classes. This is due to the loss in signal strength and mitigation of amplitude caused by the fading channel.

1-taps	2ASK	2PSK	2FSK	4ASK	4PSK	4FSK
2ASK	81	0	0	19	0	0
2PSK	0	97	2	0	0	1
2FSK	0	2	98	0	0	0
4ASK	31	0	0	69	0	0
4PSK	0	0	0	0	100	0
4FSK	0	3	0	0	0	97

Table 4.11: Contingent table under Rayleigh fading channel and thresholds

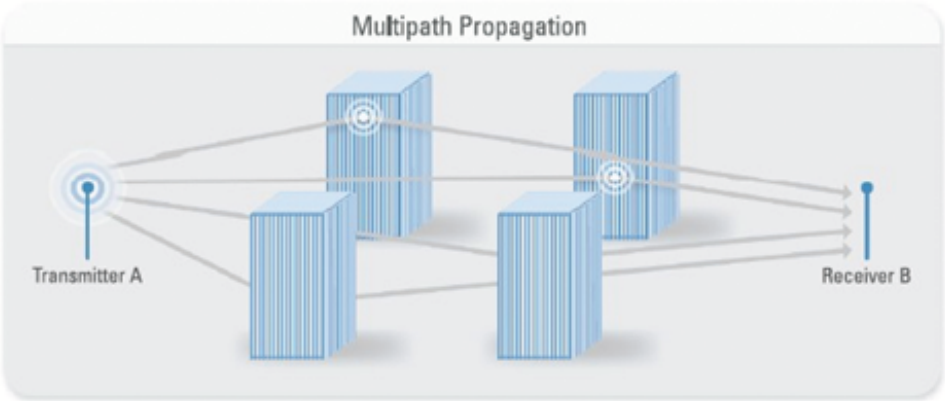

Figure 4.5: Multipath propagation in an urban environment

CHAPTER 5

TESTING AND EVALUATION OF THE PROPOSED SYSTEM

In this section, the AMI algorithm discussed earlier has been implemented on hardware using Xilinx FPGA. Furthermore, the schematic analysis, place and route analysis and device utilization statistics are evaluated and illustrated. The hardware implementation has been tested and verified for all 6 digital modulation schemes.

The developed algorithm has been simulated using Matlab v2010b and verified for different modulation schemes under Additive White Gaussian Noise (AWGN) and other channel impairments. The proposed AMI algorithm was converted into hardware descriptive language (HDL) using Verilog. The converted Verilog code was compiled, synthesized, placed & routed and implemented by Xilinx ISE webpack series 12.1. Figure 5.1 shows the block diagram illustrating the process of hardware implementation of the proposed AMI on FPGA. In the presented

implementation, Xilinx Spartan 3E - 1200E FPGA has been utilized for implementation.

5.1 Designing the Architecture

Testing simple combinational logic tasks on an FPGA can be done easily but implementing a complex sequential algorithm requires planning with great detail. Small design mistakes can lead to great amount of time wasted in rectifying the error. It is very hectic and time consuming process to identify errors in a design running on the FPGA. There are a number of variables that can go wrong if the architecture is not designed properly. A great deal of literature emphasizes on this point. Hence the design methodology followed in the proposed design is a standard practice in FPGA implementation [75] as shown in figure 5.2.

It will be shown in the following sections that how certain guidelines can be followed to ensure maximum time efficiency and correct results. There are two probable solutions in the design of the modular architecture.

1. Finite State Machine (FSM) based architecture

2. Multi-core Architecture

5.1.1 FSM based architecture

The Finite state machine based architecture is composed of two basic building blocks as also shown in figure 5.3

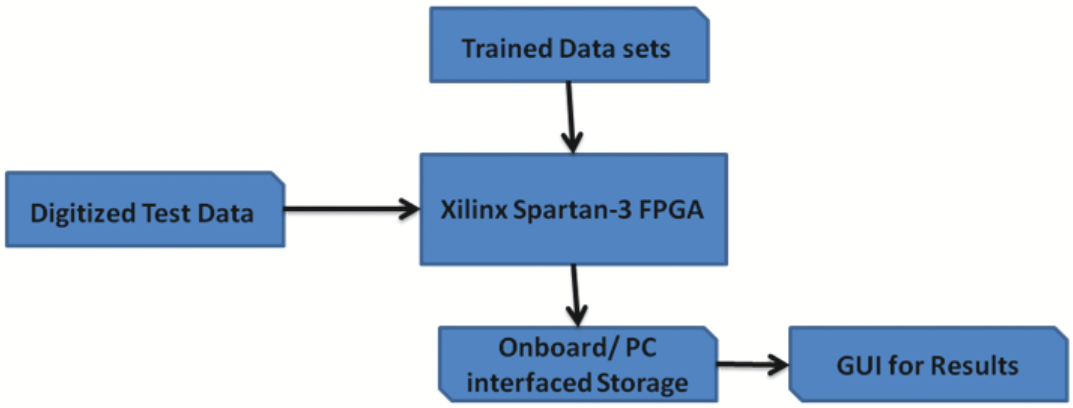

Figure 5.1: Block diagram of the implementation phase

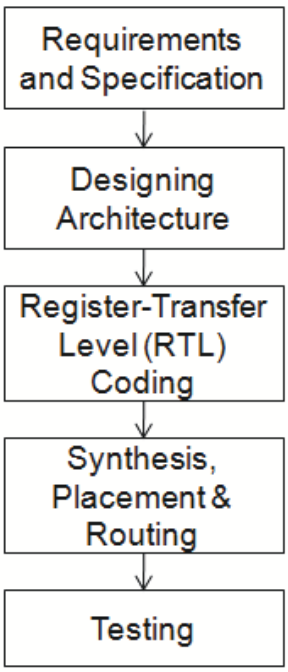

Figure 5.2: Design flow of FPGA implementation

1. Data Path

2. Control Path

The data path is the composition of the core components interlinked in a way to achieve the desired logic of the algorithm.

It was observed that the FSM based architecture is a powerful solution when a sequential algorithm is to be performed. However, it has the disadvantage of being very fragile and complex in large algorithms. A Conceptual block diagram of the data path of the designed FSM based architecture is shows in figure 5.4. The algorithm requires arithmetic operations to be performed by the floating point units and the results to be used in a recursive manner to complete the algorithm. Considering the proposed AMI algorithm, for a few number of computations the FSM based architecture worked fine if the number of states in the FSM were very low. But in the case of full scale implementation, containing over 2000 additions, 2000 multiplications and 100 subtractions and divisions, FSM architecture proved to be very cumbersome. The computations were to be stored in registers and new values were to be called from other registers. The problem arises when the registers are to be enabled and disabled and routed through multiplexers to relay information throughout the circuit. A slightest glitch in the rise time and fall time could trigger a series of errors that could only be resolved by following the computation step by step from the start every time the error was to be rectified.

In a full scale operation, the number of states for FSM architecture came close to 200 where each state had to control the entire circuit to keep it from unexpected

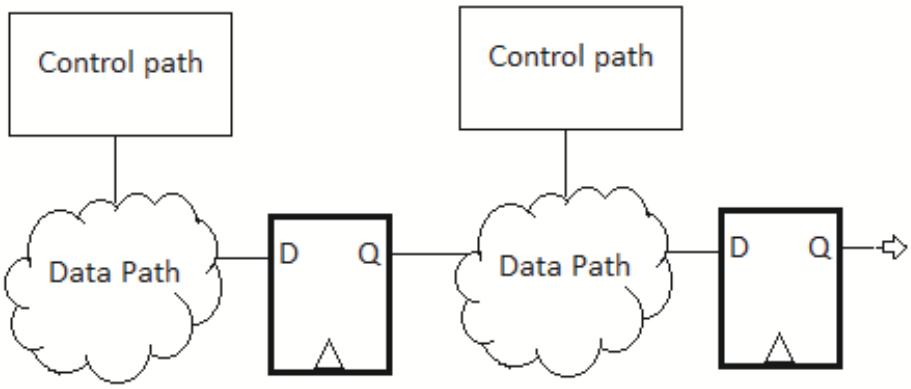

Figure 5.3: Abstract illustration of FSM based architectures

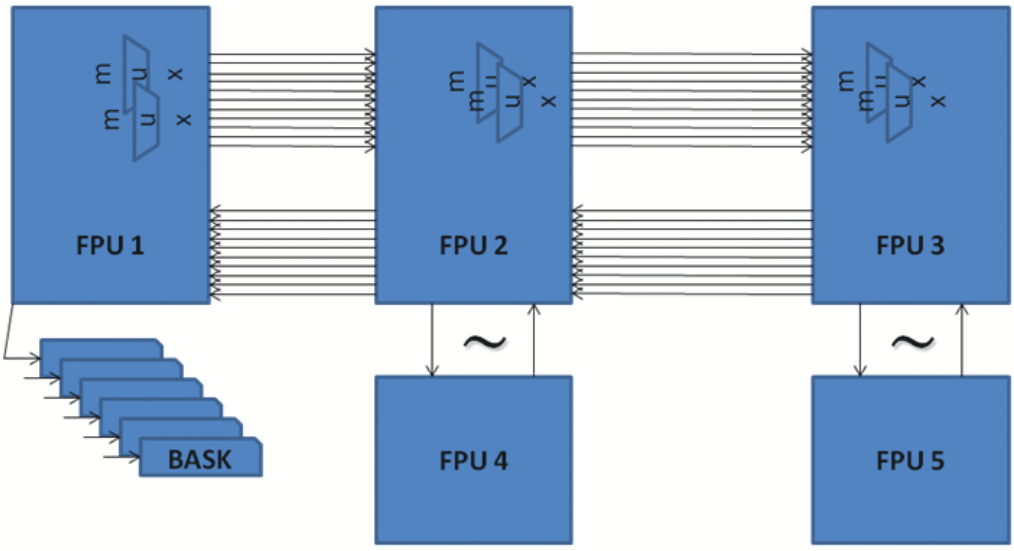

Figure 5.4: Data path of the designed FSM based architecture

behavior.

The proposed AMI algorithm was investigated on both architectures i.e. the multi-core architecture and the FSM based architecture. It was deducted that, originally the FSM based method was not designed for very large numbers of states and it should be avoided for large signal processing algorithms. With such algorithms the complexity of the hardware modules tends to grow exponentially. This growth in complexity results in a great deal of time and effort spent in verification of the proper operation. In contrast, the multi-core architecture follows a more modular design that can be scaled up or down as per the application.

Following section describes the multi-core architecture selected for the application of modulation classification.

5.1.2 Multi-core Architecture

The multi-core based architecture is the most common type of architecture that is commonly followed in almost every personal computer today. The basic components of this type of architecture consist of a computational unit, data memory, instruction memory, multiplexers and registers. The system works on a set of pre-written instructions that compute one after another and accesses memory to load and save values. All instructions are aligned in a queue that will access a central computation unit to execute the instructions one after the other. This ensures two important factors of any efficient architecture, i.e. pipelining and scalability. The proposed hardware module for automatic digital modulation identification is

shown in fig 5.1.

This system has the advantage of possibly being employed in most of the computational algorithm solving applications. Its generic design and adaptability gives the architecture great deal of possibilities. It was found that by developing a multi-core architecture some prominent advantages could be achieved as compared to the FSM based architecture

1. Low complexity architecture even when used with large designs
2. Easier to debug as compared to the FSM based architecture
3. Non-accumulative glitches, hence reducing the verification time.

The multi-execution architecture has the same disadvantage that an FPGA has as compared to an ASIC. Since the architecture provides great deal of flexibility it also taxes the computation with operational overheads. The multi-execution architecture is shown in the figure 5.5 below.

Other advantages of using a multi-core architecture are as following.

1. **Pipelined Architecture**

   The design includes an instruction memory and multiple computational cores embedded in one design such that all the instructions compute one after the other and this creates a pipelined architecture. All the instructions are written to compute according to the algorithm and each instruction is part of the bigger equation that is being solved on the computational core.

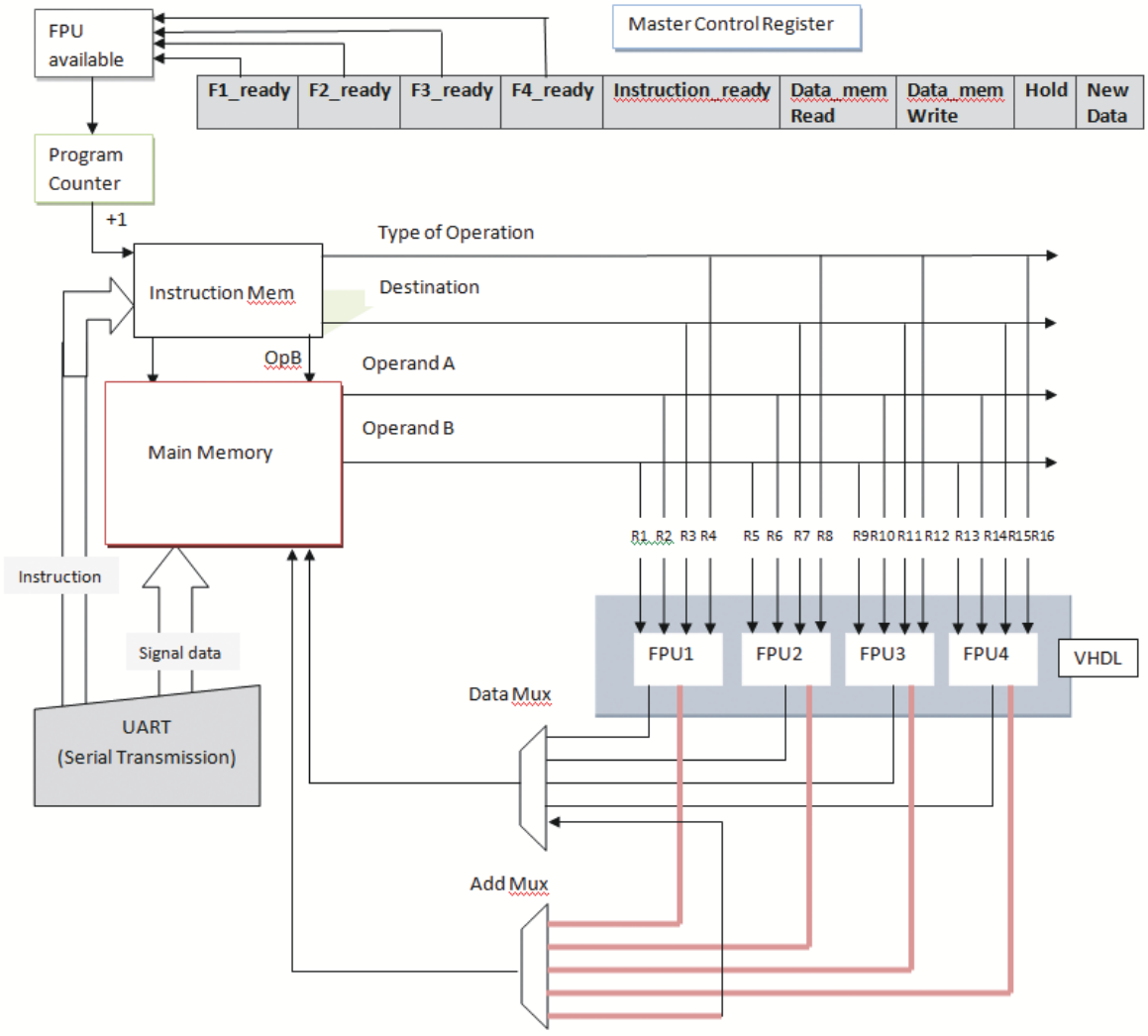

Figure 5.5: Proposed Multi-execution architecture

2. **Parallel Processing**

   The proposed architecture is capable of performing multiple computations at the same time due the presence of multi-execution floating point unit (FPU) cores in the design. All the FPU cores are controlled by the master control register in accessing the data bus. One after the other they can be assigned operations that they can compute at the same time.

3. **Scalable Hardware**

   The biggest advantage of the system is that the architecture can be scaled according the computational needs and or the resource bearing of the platform. In the current FPGA we can accommodate up to 8 floating point unit cores in the architecture. All FPUs will be in a pipelined and parallel system that can retrieve instructions and start computing simultaneously. Very little changes have to be made to scale up or down the system. The system can also be modeled to adapt dynamically to application requirements, such that the architecture adds or removes computational cores if needed.

5.2 Components of the Architecture

The following section describes each component used in the proposed architecture in detail.

5.2.1 Dual Port Instruction Ram

All of the memory components were generated from the Xilinx Core Gen tool shipped free with the ISE webpack 12.1 software. The instruction ram contains the instructions that the proposed architecture will perform once it is signaled to start. The structure of our instruction memory is illustrated in figure 5.6. This requires 38 bits of data to be stored in one single location of the Instruction memory.

Since our algorithm requires around 1000 additions and 2000 multiplications, we require a total depth of our instruction ram to be 4096 locations. The generated memory module by core gen is depicted in figure 5.7.

The RAM module shown in figure 5.7 is a true dual port ram such that it has a multiplexing circuit built inside the memory to read and write at the same time. This extra functionality was required to make the system 100% real time and perform operations in parallel.

5.2.2 Dual port & Dual module Data RAM

The design includes two dual port memories that duplicate the data being stored in them as shown in figure 5.8. This duplication of ram was required to read both operands at the same time from the memory and save the overhead and additional hardware required to get a sequential retrieval.

Structure of the data stored in the data memory is shown in figure 5.9. The operand to be stored in memory is 32 bits long, since the IEEE-754 standard

Operand A (11)	Operand B (11)	OP (3)	Destination (11)	Instruction add (13)	Free (5)	Control (8)

Figure 5.6: Structure of the Instruction RAM elements

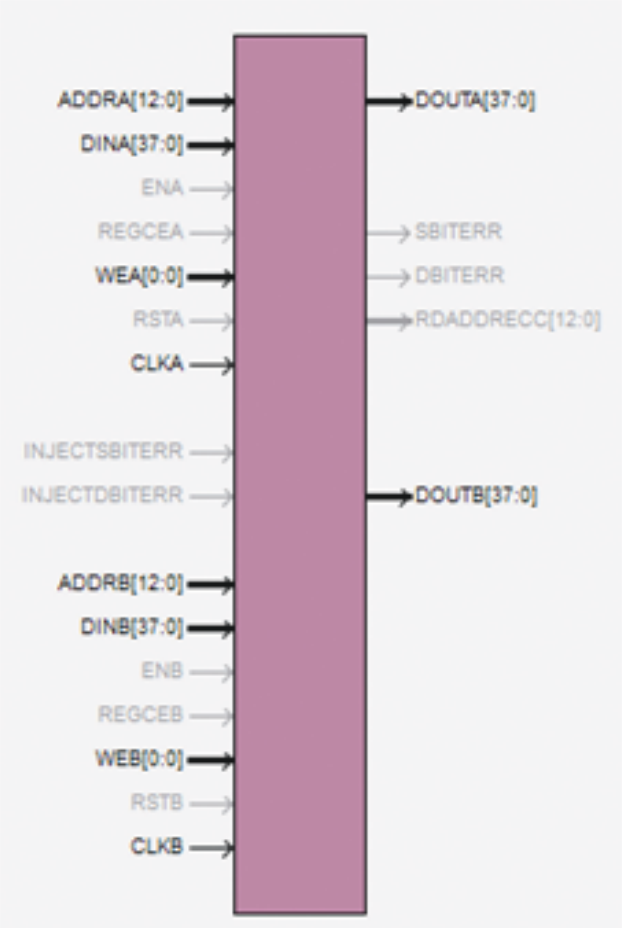

Figure 5.7: Block diagram of the true dual port instruction memory

converts each integer into a 32 bit operand.

5.2.3 Master Control Register

The master control register can be considered as the flagship of all status signals, see figure 5.10. The system reads instructions from the instruction memory. According to the instruction, data is extracted from the data memory and the computations are performed in the Floating point units (FPU). When one FPU is busy doing the computation, the status flag for FPU 1 is changed in the master control register and this FPU will not be accessed until the ready flag is 1 again. This follows for all four FPUs in the system.

There can be several floating point unit FPUs aligned next to one another providing the parallel computational ability and sharing the same bus to receive data. The bus will also be controlled by the master control register, allowing only one FPU, at a time, to access the data bus.

As one FPU finishes an operation, the results have to be stored in the data memory. The result bus is also shared between all the FPUs but controlled by the master control register to multiplex the outputs of the FPUs into one data memory. The hold flag is signaled when a recursive equation is being solved and further computation depends on the result of the current operation. In this scenario no new FPU is assigned any job until the value required for the recursive operation is obtained.

The 'instruction ready' flag calls the FPU to obtain data of the next instruction

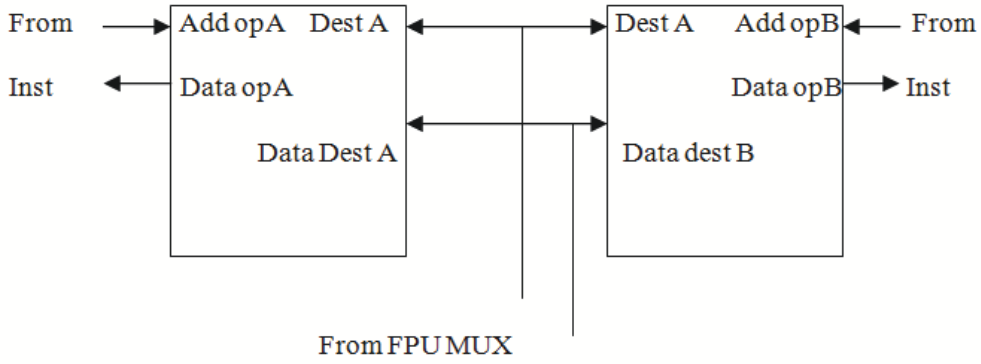

Figure 5.8: Design of the Duplicated true dual port data ram

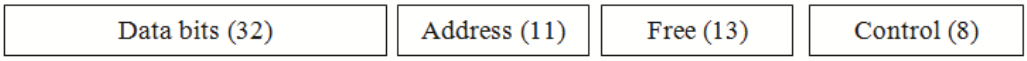

Figure 5.9: Structure of the Data Ram elements

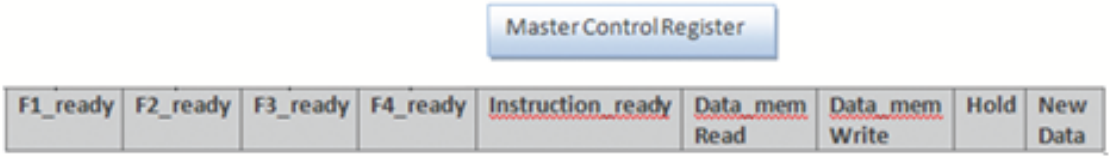

Figure 5.10: Structure of the Master control register

Operation	Number of Clock Cycles
Addition	7
Subtraction	7
Multiplication	12
Division	35
Square-root	35

Table 5.1: Clock cycle requirement for available arithmetic operations

	FPU#1 (Presented here)	FPU #2 (Usselmann)
No. of logic elements	*3468	7392
fmax	7	6.17 Mhz
	Clock Cycles	
Addition/Subtraction	7	3
Multiplication	12	3
Division	35	3
Square-root	35	NA

Table 5.2: Performance Comparison to competing FPU by Usselmann

in queue. The 'instruction ready' flag goes high only when the instruction has been read and all the required components have been obtained and put on the bus.

5.2.4 Floating point Unit (FPU)

The concept of a floating point unit is to address the arithmetic calculation using IEEE 754 standard [98]. The Floating point unit module employed here was developed by Jidan Al-Eryani. The FPU core architecture is shown in figure 5.11.

Operation cycles of the FPU Core

This FPU uses pipelining to achieve the fairly high throughput. Table 5.1 below describes the operation cycles for each arithmetic operation.

Comparison to FPU by Usslemann

The FPU core by Al-Eryani was preferred over the highly popular FPU core pre-

sented by Usselmann [103]. This is because the maximum achievable frequency of FPU by Al-Eryani was roughly 16 times higher as compared to that of Usselmann. Also the numbers of Logic blocks used were roughly half than that of [103]. A comparison of the two FPUs is illustrated in table 5.2.

5.2.5 Mixed Signal Implementation

The IEEE 754 standard FPU by Al-Eryani was written in VHDL language while our designated HDL design language was chosen to be Verilog. The difference in languages posed a problem of incompatibility with many compilers. Two possible solutions were tested

1. Convert the VHDL using automatic translators

   The usage of automatic language translators always produces a resultant code that is mostly incomprehensible and un-modifiable. Manual translation is a challenging task in itself and an in-efficient practice.

2. Mixed Signal Compilers and Synthesizers

   In order to compile both VHDL and Verilog together in the same program we used Modelsim 6.5e together with Xilinx ISE 12.1 design suite. The combination of these tools provide mixed signal compilation and synthesis.

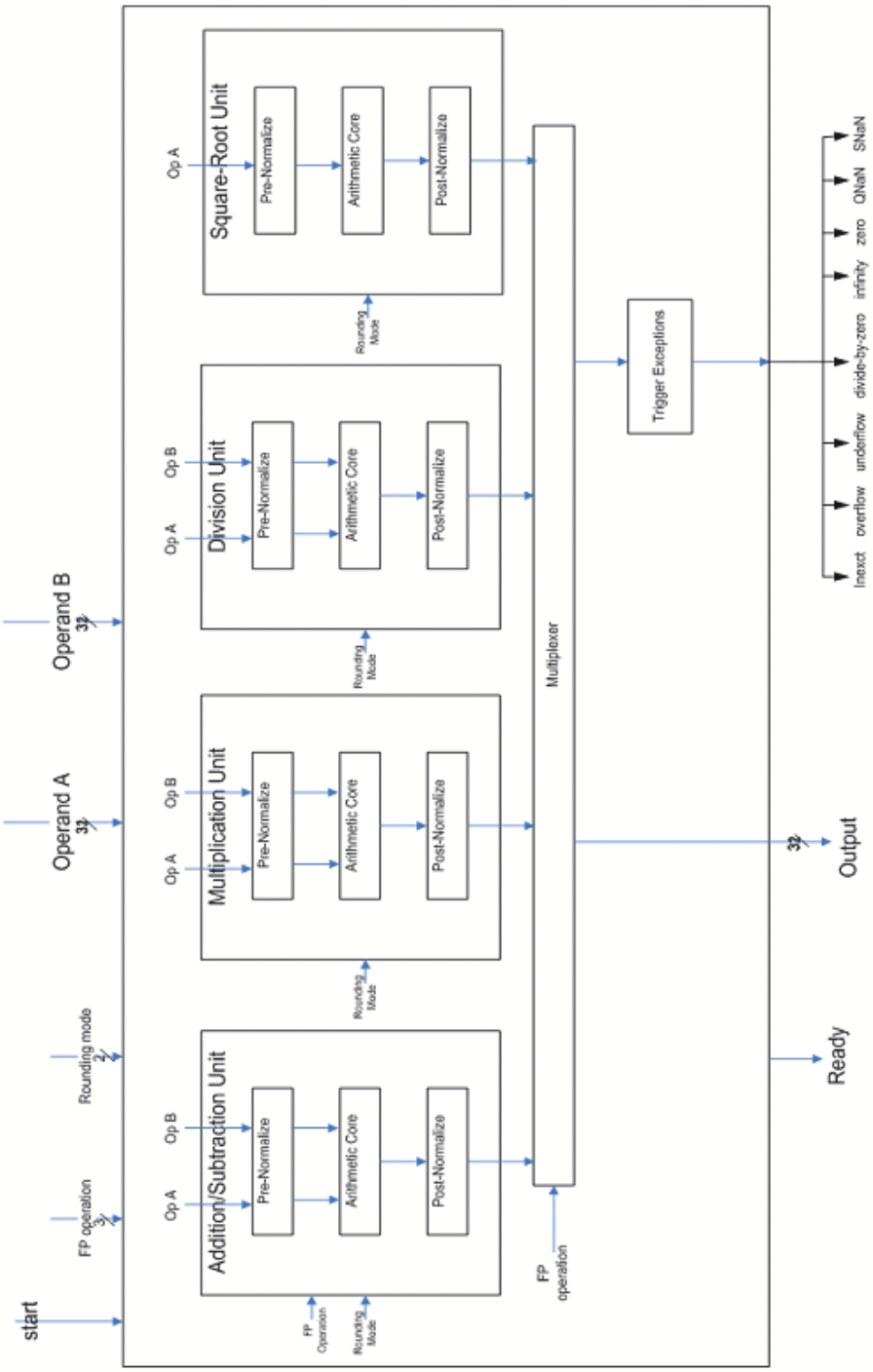

Figure 5.11: The FPU architecture by Al-Eryani

5.3 Instruction set

Every microprocessor and microcontroller has a set of pre-defined instructions developed by the designer. These instructions enable the programmer to develop applications and programs to construct a logic operation.

In reprogrammable platforms, such as FPGA's and ASIC's, the user has to develop his own architecture. This gives the option to the user to write custom instruction sets. In the case of implementing DSP algorithms on hardware, the user has to develop instructions that will perform the required DSP algorithms on the FPGA.

The proposed architecture for modulation classification is a generic mutli-execution core that can implement a very wide range of algorithms. The goal is to develop an architecture that is not specific to only one algorithm, but can implement any relative algorithm. With this objective, the proposed architecture contains the following instruction set as shown in figure 5.12.

5.3.1 Increment

The increment instruction is almost present in all of the instruction based architectures. All instructions are executed in a sequential manner. Thus the need arises for a dedicated hardware that can move from one instruction to the other. The increment hardware created for the proposed architecture is shown in figure 5.13. As the increment command 'INC' is given as an instruction, the hardware adds 1 to the current value of the program counter. The incremented value of the

program counter then points to the next instruction in the instruction list.

5.3.2 Loop

Most of the digital signal processing algorithms works on discrete signal data. The methods take each sample of the data and apply a series of operations on each of them. Almost all of the time the real world discretized data is a long series of numbers. In terms of hardware, this means having a dedicated module for looping the program counter 'N' number of times. Where N is the number of sampled data points.

The proposed architecture contains a dedicated instruction of loop with the op-code '11101' as shown in figure 5.14. Other parts of the instructions contain 'destination', and the 'loop-count' fields, also called arguments. The 'destination' argument sets the starting point of the program counter and then the program counter is reset. Every time a program counter reset has been performed, 1 value is deducted from the 'loop-count' value. Thus the loop instruction repeats itself 'loop-count' number of times. The schematic block diagrams of the loop instruction are shown in figure 5.14 and 5.15.

5.3.3 Zero Crossing Count

The proposed AMI algorithm requires certain thresholding steps after the classifier has finished the processing. The zero crossing count was selected as the thresholding parameter to differentiate between certain classes. The details of the

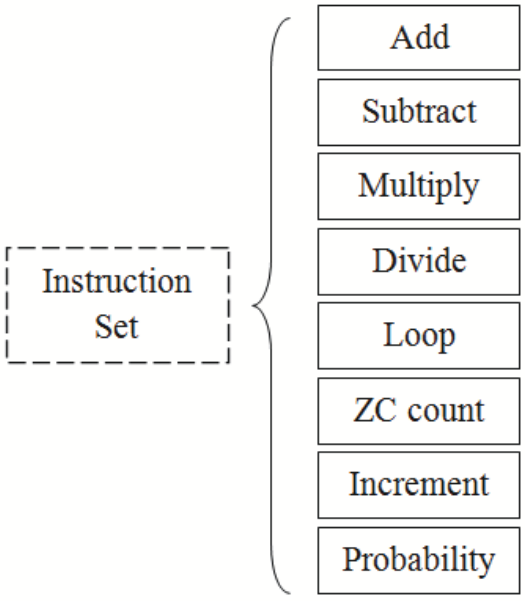

Figure 5.12: Different instructions developed for the proposed architecture

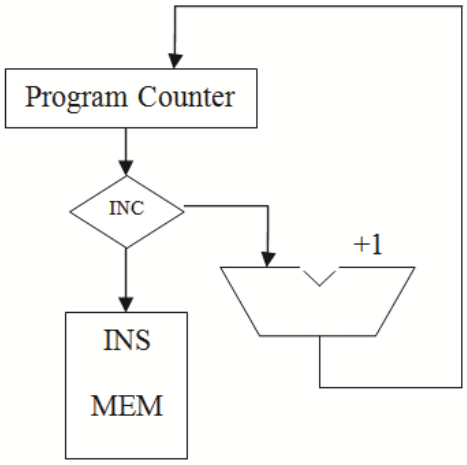

Figure 5.13: Hardware designed for the increment instruction

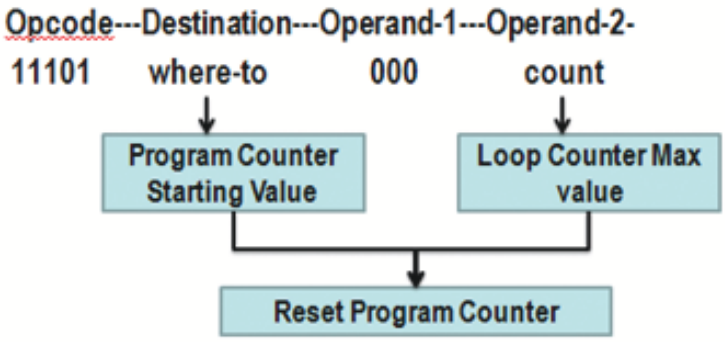

Figure 5.14: Block diagram of describing the Loop instruction

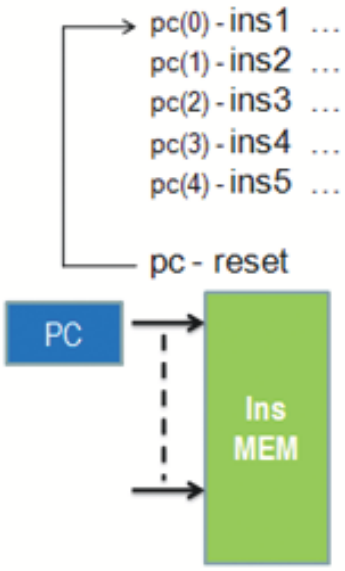

Figure 5.15: Loop reset condition and direct manipulation of the program counter

Si	Si+1	0 ▶ 0	0 ▶ 1	1 ▶ 0	1 ▶ 1
		No-OP	Sum=Sum+1	Sum=Sum+1	No-OP

Table 5.3: Conditions for Zero crossing count

operation and requirements of the ZC count were given in the previous sections. In terms of hardware implementation, the ZC is the toggle of the sign bit. Since the proposed architecture used IEEE-754 floating point number representation, only the 32nd bit is to be monitored in order to identify the zero crossing. Table 5.3 depicts the conditions when the 32nd bit is toggled and the zero crossing would be counted.

The ZC count operation is shown in figure 5.16 below.

5.3.4 Probability comparison

After the completion of the proposed modulation classification algorithm, the final probabilities are compared to find the closest match. Since the algorithm uses one-vs-all (OVA) methodology, all probabilities are individual. Thus the probability of each class should be compared to the rest of the classes. This idea is illustrated in figure 5.17 below

After the probabilities of the six classes are evaluated, they are compared to each other. This task is performed using a lookup table (LUT) as shown below.

casez(sign-reg)

16'bzzzzzzzzzzzz00000: LED= 6'b000001

16'bzzzzzzz0000zzzz1: LED = 6'b000010

16'b1zzzz00zzz1zzz1z: LED = 6'b000100

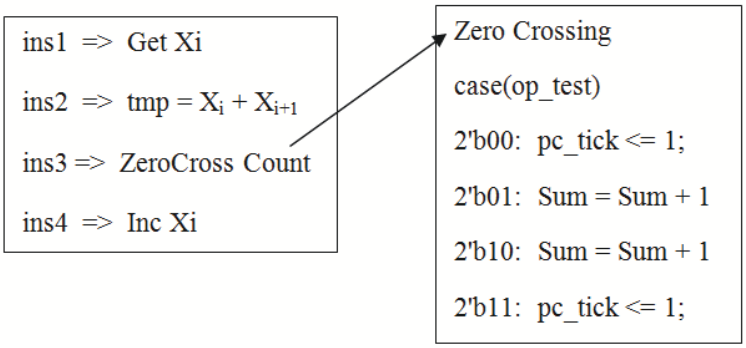

Figure 5.16: Sample instruction describing the zero crossing operation

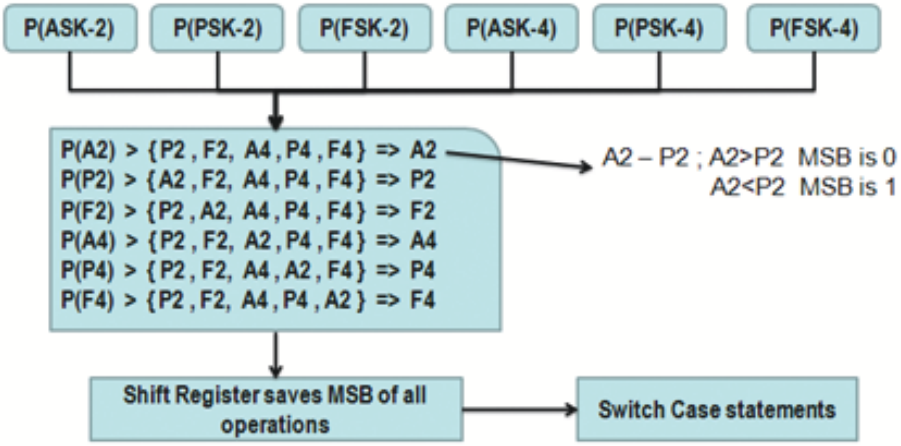

Figure 5.17: Block diagram of the final probability comparison

16'bzz00zz1zz1zzz1zz: LED = 6'b001000

16'bz0z1z1zz1zzz1zzz: LED = 6'b010000

16'b011zzzz1zzz1zzzz: LED = 6'b100000

default: sseg-shift = 6'b101010

endcase

The use of a switch-case statement saves a lot of computational overhead. The if-else statements were not used due to their complex implementation modules.

5.4 Target Device Parameters

The device used for implementation was the Nexys-2 FPGA development kit by Digilent inc [104]. The Nexys2 board houses a Spartan 3E - 1200E Xilinx FPGA. Figure below shows the block diagram and features of the Nexys2 development board.

The Nexys2 board has a 50MHz oscillator for clock generation. The clock can also be imported from an external source. A ST3232 RS232 voltage converter is also present for serial port interfacing. A UART was developed on the Nexys 2 FPGA board to receive data streams from a personal computer. The UART works on 19200 baud rate with 8 data bits. The system was tested using an usb-to-serial port cable to verify the functionality.

Although the Nexys2 kit provides an onboard 16 Mbytes of SDRam and 16Mbytes of Flash memory, but the on-FPGA memory was utilized for low overheads and reduced area usage.

The Spartan-3E architecture includes dedicated block memories of 4096 bits each, clock DLLs for clock-distribution delay compensation, clock domain control and 3-State buffers associated with each CLB. Values stored in static memory cells control the configurable logic elements and interconnect resources [104]. These values can be loaded into the memory cells on power-up, and it can be reloaded if change in the function of the device is necessary.

Following is the summary of the Xilinx Spartan 3E FPGA. The FPGA has 62.6 Kbytes of RAM available that can be used a scratch ram or as a buffering space. The proposed architecture design had continuous logging and processing of data stream that enabled 95% usage of the Block RAMs on the FPGA. Other details are provided in Table 5.4 below.

Table 5.4: Summary of the Spartan-3E FPGA attributes

Device	XC3S1200E
System Gates	1200K
Logic Cells	19,512
CLB Rows	60
CLB Columns	46
Total CLBs	2,168
Total Slices	8,672
Distributed RAM bits	136K
Block Ram bits	504K
Dedicated Multipliers	28
DCMs	8
Maximum User I/O	304
Maximum Differential I/O pairs	124

Although the usage of FPGA memory left no space for the buffer or scratch pad RAM, for the buffer, but enabled to skip the hectic process of memory interfacing and the overhead generated by it. Also some area was further reduced by not

employing the memory interfacing modules.

5.5 Algorithm constraints

Every application has its own set of requirements that put certain constraints on the hardware. The architecture has to be modeled to cater for these constraints. The proposed modulation classification algorithm puts certain constraints in designing the architecture in its hardware implementation as discussed below

5.5.1 Memory Requirement

The memory constraint placed by the AMI algorithm is the requirement of continuous access to the sampled waveform. This translates into a massive overhead in accessing the memory each time the system has to perform any computation. The waveform is sampled into a data set of 1x1000 data points that are required for every computation. Also, the same data has to be used multiple times in the procession of the algorithm. This means that the new resultant values from the computations cannot be saved on top of the data and it cannot be overwritten to save memory data space. The result is larger memory modules that take up much more space and simple buffers or registers cannot be used for storing the data. A dedicated memory has to be initialized.

5.5.2 Recursive Equations

The AMI uses Bayesian approach as a main classification model and extracts features to feed in the classifier. When extracting the features, the equations of the feature parameters are recursive in nature. More specifically, when calculating the four moments, we need to compute a number of parameters first in order to finally evaluate the moments. This places a constraint of carefully sequencing the instructions such that each recursive computation is performed after its pre-requisites have been fulfilled. Also, in the master control register there is an additional flag specially designed to halt all operations and only compute the bottleneck parameter before moving forward.

5.6 Synthesis & Implementation

The first stage of the synthesis is to analyze the developed Verilog code to check the compatibility for synthesizing. After analyzing the source code, the target device has to be synthesized and the netlist to be created. The device utilization summary for the implementation on Spartan3E-XC1200-fg320 is shown in Table 5.5.

It can be observed from Table 5.5 that the proposed architecture utilizes 32% of bonded Input-Output Banks (IOBs). Table 5.5 indicates that a minimum of 82 IOBs are needed for implementing the proposed algorithm on a FPGA. It may be seen from Table 5.5 that the available Spartan-3E is providing max of 250 IOBs. The device utilization clearly indicates that the selected target device S3E-

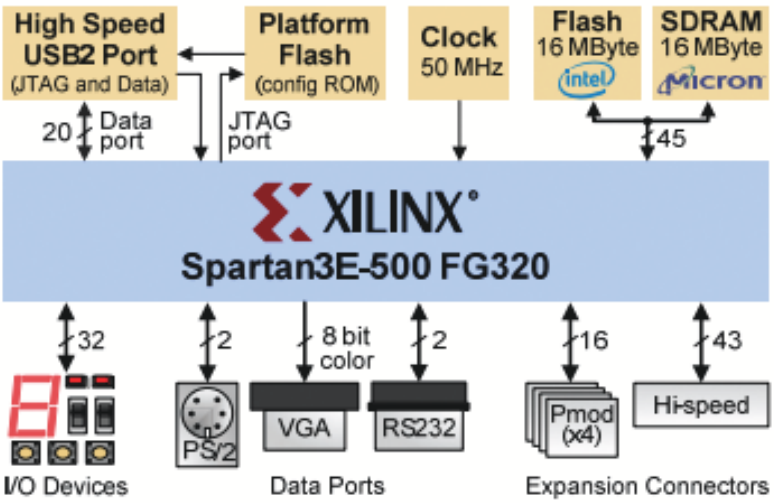

Figure 5.18: Block diagram of the Nexys2 development kit

Name of Block	Available	Utilized	% of Utilization
Number of Slices	8672	5049	58%
Number of Slice Flip FLops	17,344	2565	14%
Number of 4 input LUTs	17244	9433	54%
Number of bonded IOBs	250	82	32%
Number of GCLKs	24	2	8%

Table 5.5: Device Utilization of the proposed FPGA architecture

1200efg320 is well suited to implement the developed model. The summary of the timing report during the synthesis process is illustrated in Table 5.6.

Table 5.6 shows that the minimum time required for 1 clock cycle is measured as 50.346ns. More specifically 27.934ns logic and 22.809ns route, that has a percentage of 55.0% and 45.0% for logic and route respectively. The total memory required to implement the proposed method has been observed as 358348 kilobytes.

A top level modular structure of the architecture is shown in figure 5.20. The top level structure is composed of a two serial transmission lines, 7-seg and LED display pins, clk, reset and the functional button pins. The device level schematic for the generated model is shown in Fig 5.21. The device level schematic shows the LUT utilization per CLB for the designed architecture. The highlighted orange squares display the LUT utilization.

The synthesized netlist describes the interconnection of blocks, the logic cells within the blocks and the logic cell connections. The netlist is fed for floor planning before placing and routing process. Floor planning is a process that translates the logical description of the HDL to the physical description i.e. modules to gates. Another task of the floor planning operation is to put the functional blocks on the locations of I/O pads and handle clock distribution. The floor planning for the proposed model has been generated and the top view of the same is illustrated in Figure 5.22.

The floor plan is divided into 4 blocks of different colors. These four colored

Parameter	Duration(ns)
Minimum Period	50.346
Minimum input arrival time before clock	9.273
Maximum output required time after	5.909
Maximum combinational path delay	No path found

Table 5.6: Timing Report for Spartan 3E-1200e FG320

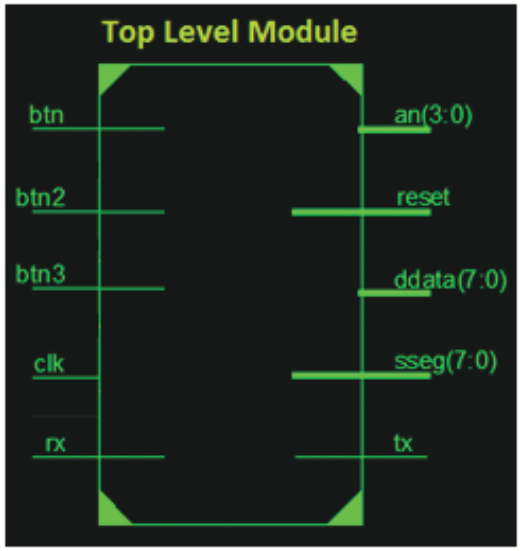

Figure 5.19: Top level block level schematic of the proposed architecture

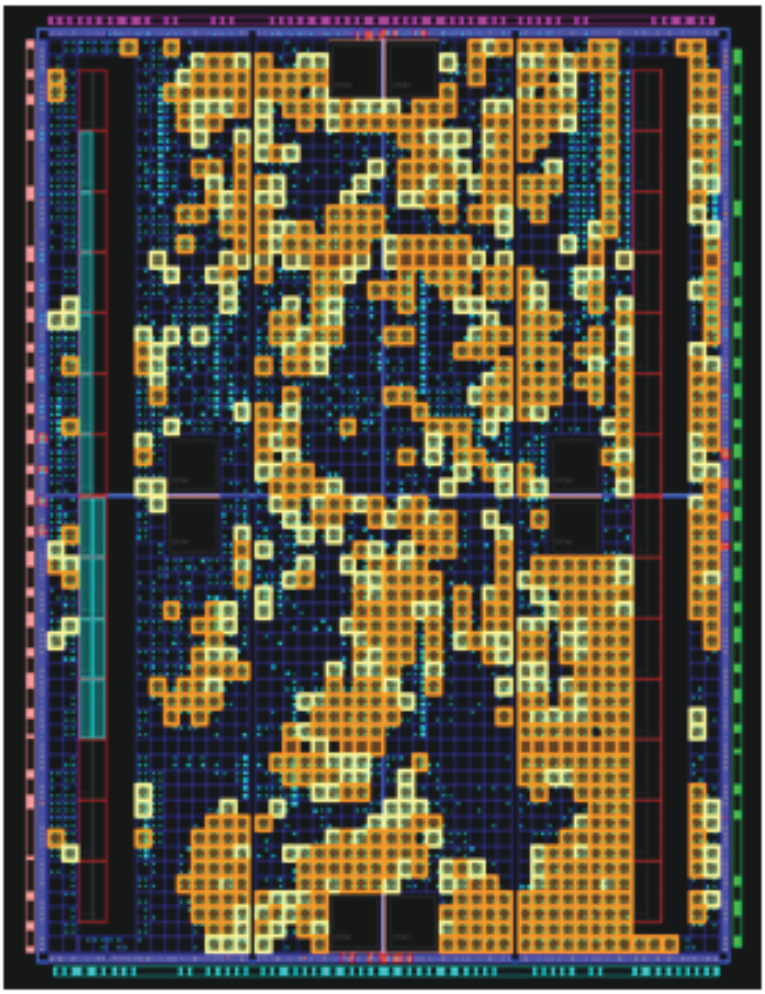

Figure 5.20: Device level schematic for the generated model

areas display the four regions of IOBs. The grey circles depict unused I/O banks. The inside black region shows actual field programmable gate arrays. The green square surrounding the black FPGA are the package pins for I/O communication. It can be summarized that the area required to implement the proposed architecture for the target chip S3E-1200efg320 is 54% of the actual area which indicates the expansion possibilities for various demodulation schemes in future.

5.6.1 Power Utilization

The power required for execution of modulation identification process using the target device S3E-1200efg320 has been computed and listed in Table 5.7 for individual components. The total dynamic power for entire module has been calculated using the relation [80].

$$P = sum_{i=1}^{25825} P_{ci} + \sum_{j=1}^{83} P_{ioj} + P_{clk} \tag{5.1}$$

Where

P is total Dynamic Power

P_{ci} is the power of the ith component

P_{ioj} is the power of jth IO pad

P_{clk} is clock power

The total power required to implement the proposed method is the combina-

Name	Power(W)	Quantity Available	Quantity Used	Utilization(%)
Clocks	0.007		1	
IOs	0.013	250	82	32%
Logic	0.010	17244	9433	54%
Quiescent	0.155			
Dynamic	0.037			
Total Power	0.193			

Table 5.7: Power required for the target

tion of Quiescent power and dynamic power. It is seen from the Table 5.7 that the total power required to implement the proposed module in Virtex-E S3E-1200efg320 is 193 mW.

5.6.2 Verification of the hardware

The implemented AMI reconfigurable digital modulation identifier has been verified and validated for different digital modulation schemes such as 2-ASK, 4ASK, 2-PSK, 4-PSK, 2-FSK, and 4-FSK. The developed architecture has been tested for various modulation schemes under Rayleigh fading channel conditions and AWGN noisy environment for different SNR. The verification and validation of the proposed model has been carried out for different digital modulation scheme and authenticity of the implemented hardware has been analyzed. The percentage of correct identification of the proposed AMI hardware at 10 dB SNR has been estimated and shown in Table 5.8.

5.6.3 Design Summary

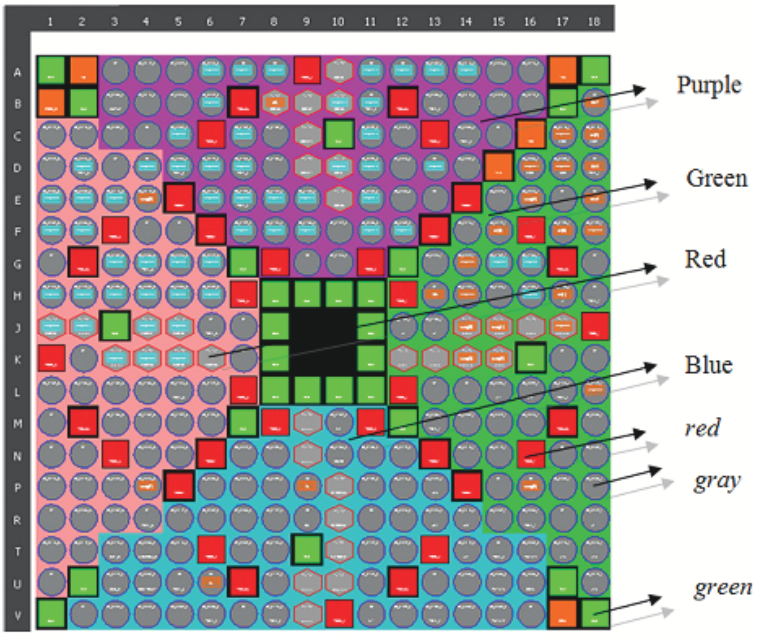

Figure 5.21: The floor planning for the proposed model

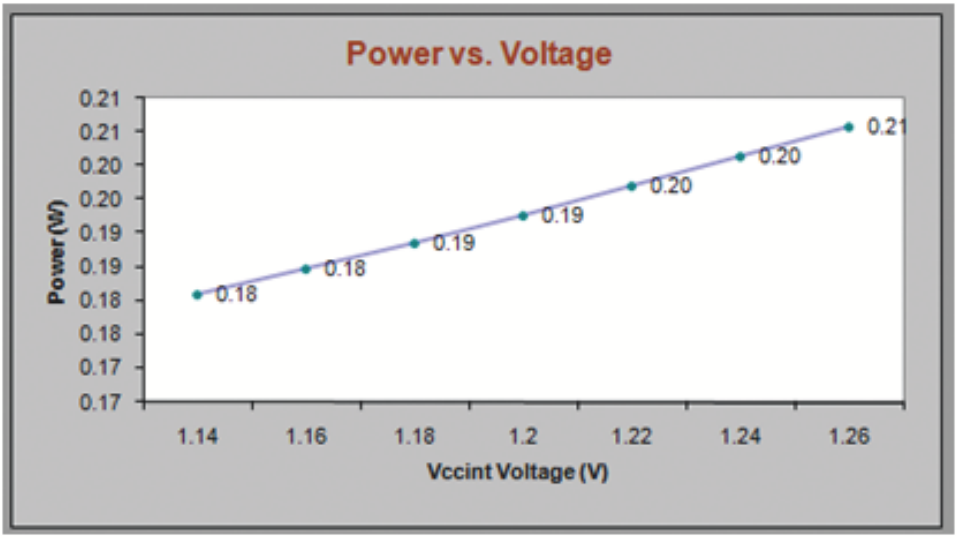

Figure 5.22: Power vs Voltage curve of the proposed architecture

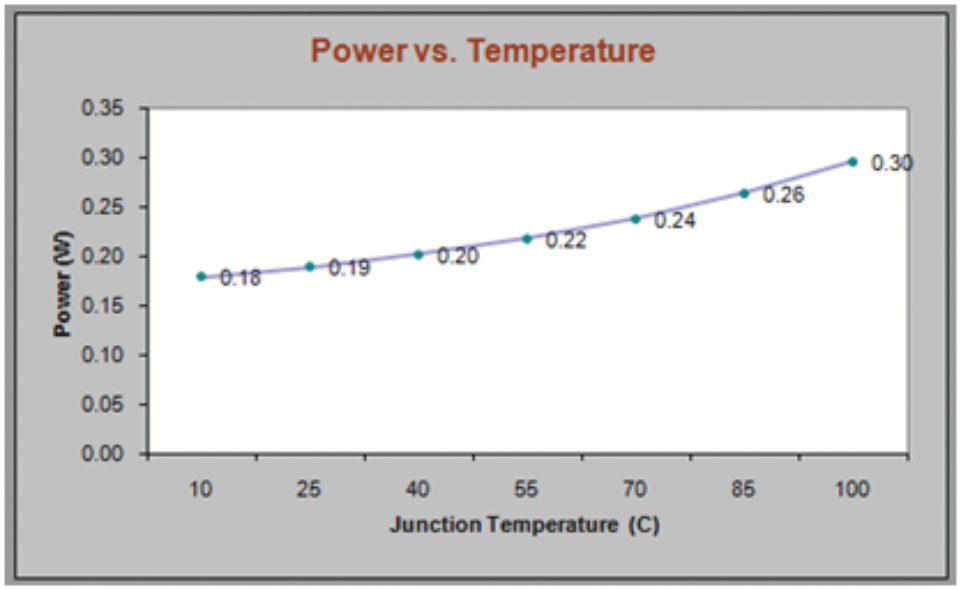

Figure 5.23: Power vs Temperature curve of the proposed architecture

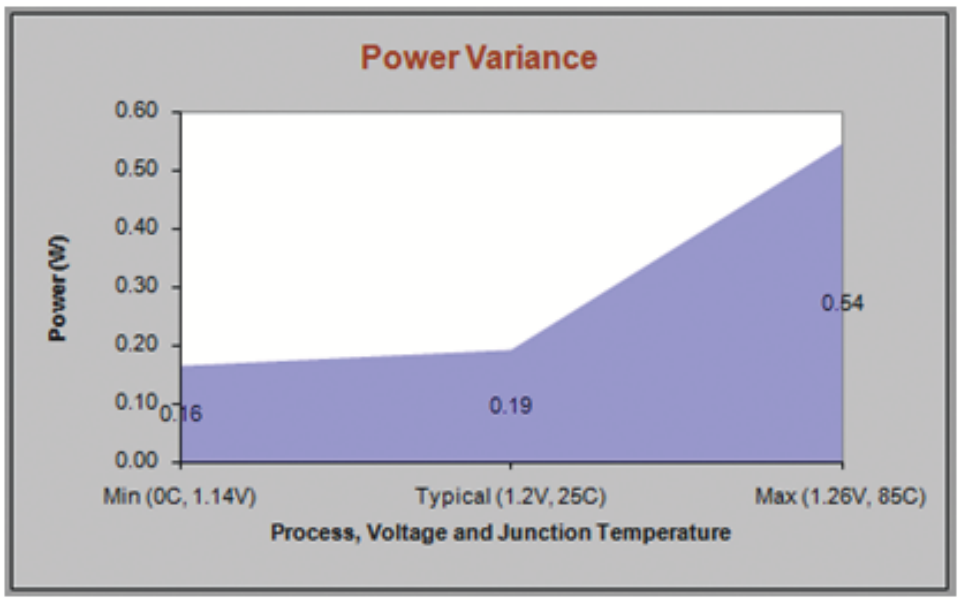

Figure 5.24: Power variance of the proposed architecture

Modulation Scheme	% of identification
2-ASK	100
2-PSK	97
2-FSK	100
4-ASK	100
4-PSK	99
4-FSK	100

Table 5.8: Percentage of identification of the proposed modulation classifier at 10dB SNR

Table 5.9: Summary of the Spartan-3E FPGA attributes

	Port A	Port B
BRAMs	10	10
Toggle Rate	12.5%	12.5%
Clock (MHz)	20	20
Enable Rate	25%	25%
Bit Width	1	1
Write Rate	50%	50%
Utilzation	35.7%	35.7%

Table 5.10: Summary of the Spartan-3E FPGA attributes

Clock (MHz)	20
LUTs	9371
Shift Registers	62
Select RAMs)	0
Flip FLops	67
Toggle Rate	12.5%
Average Fanout	4
Multipliers	4
Power (W)	0.010

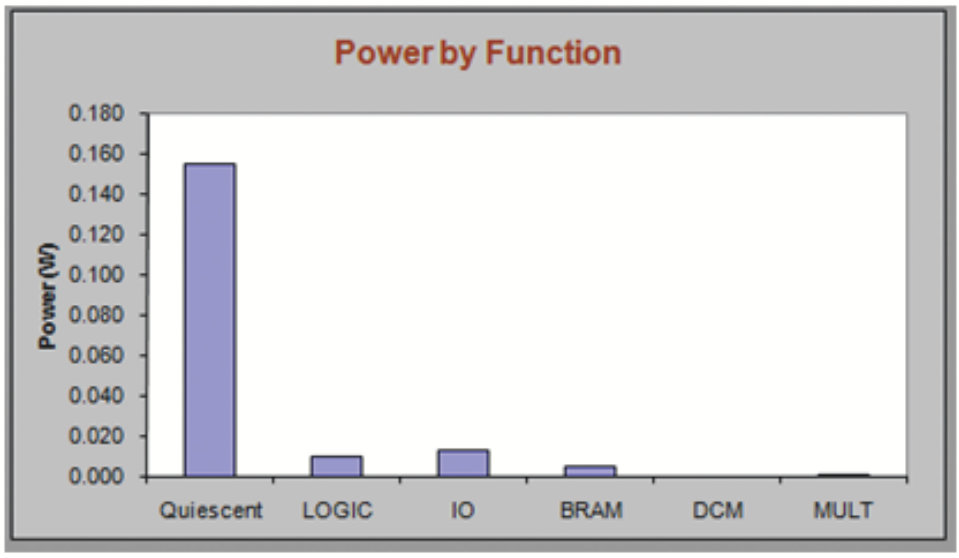

Figure 5.25: The power division with respect to function

CHAPTER 6

DEVELOPMENT OF THE GUI

The testing of the proposed AMI algorithm on FPGA was performed offline. The sampled waveform was transferred from a PC to the FPGA board for verification. Hence a graphical user interface (GUI) was developed that tests and verifies the algorithm by downloading data and controlling the FPGA. This GUI has two major components that are, serial interfacing and software development. The details of these components are given in the following sections.

6.1 Serial Interfacing

The proposed algorithm works on signal waveforms that are required to be sampled and transferred to the FPGA. There can be many techniques by which the sampled data can be transferred to the FPGA depending upon the hardware support. For example, in the laboratory for testing purposes this interface can be composed of a function generator coupled with a USB port, Parallel port, custom bus, or a serial port. The serial port is used in the proposed framework because it is

relatively very easy to setup and control. Serial port interfacing is usually not recommended in the final realization due to its slow speed and requirement for additional hardware modules. But alternate options like USB port or a custom developed bus can be very challenging and time consuming tasks. Since the goal is only testing and verifying the AMI algorithm, the PC-FPGA interface utilizes serial interfacing modules.

Universal Asynchronous Receiver / Transmitter (UART) A universal asynchronous receiver is a parallel to serial line converter that sends and receives data in serial and provides the data in parallel to the FPGA. A UART is required when transferring data from a serial port to the FPGA because the voltage level of a PC-serial port is + / - 12 volts while the TTL logic in FPGA uses 0-5volts. This conversion is necessary otherwise the circuit will be damaged. The voltage converter cannot be built on the FPGA using combinational logic. This voltage converter hardware should be implemented as a standalone IC [105]. The Nexys2 development kit has a standard RS232 to TTL voltage converter available on-board. The RS232 to TTL converter IC was utilized in the UART module.
A serial port can transmit 6, 7 or 8 bits at a time with different varieties of start and stop bits as shown in figure 6.1. Data transmission starts with logic 0 and terminates with logic 1. In between are the data bits and the parity bit that is optional. The stop bits can be selected from 1, 1.5 or 2 bits.

In serial transmission, the transceiver system is first set upon a few parameters that overlay the guidelines of the transmission and reception. These parameters

are, as previously discussed, number of data bits, parity bit, number of stop bits and one more parameter baud rate that describes the speed of the overall data exchange. Baud rate can be selected from 2400, 4800, 9600 or 19200 bauds.

6.1.1 UART Receiver (Rx)

The basic components or UART receiver are

1. A circuit that uses oversampling to receive data
2. A circuit that creates sampling ticks
3. A circuit that acts as an intermediate buffer stage between the UART and the FPGA

An abstract block diagram of the receiver is shown in figure 6.2

6.1.2 UART Rx Finite State Machine (FSM)

Since a UART system follows a sequence of finite number of procedures, we require a finite state machine to control the flow of procedures [105]. The execution of operations in each step of the receiver are depicted in figure 6.3.
The steps followed by the state machine are explained below

1. The system will wait until an initial zero '0' bit that signals the start of operation.
2. If the sampling counter is not started then repeat the 2nd stage until the counter starts

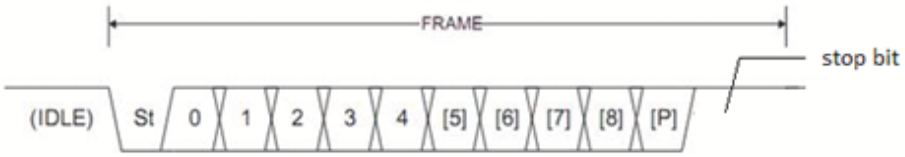

Figure 6.1: Frame of the serially transmitted data

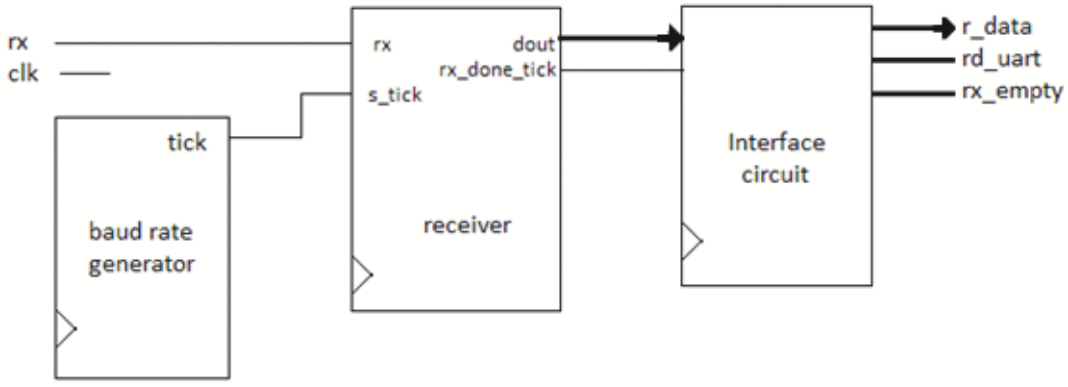

Figure 6.2: UART receiver system

3. After the start bit is 0 the sampling counter is started that counts up to 7, keep adding 1 to the counter until 7 is reached. The 7th count indicates that the start bit has reached its middle.

4. After the count is completed move on to the 3rd stage and start again the counting process, this time up to 15.

5. If the counter tick is not started go to the 3rd stage again and keep looping until the flag is high.

6. Keep adding 1 until the counter reaches 15. 15th count indicates that the first data bit has reached its middle.

7. Save the first bit and keep repeating 3rd stage until all the bits in the serial transmission are received. That means 8 data bits, 0 parity bits and 1 stop bit. The combination of data, parity and stop bits is one particular combination selected by the user.

8. The FSM then advances to the 4th stage after all the bits have been saved and assert a done flag high and going back to the idle first stage.

6.1.3 UART Transmitter (Tx)

A UART transmitter is almost same as the UART receiver. The major components stay the same, but the flags are now controlled by the FPGA. The FPGA signals a flag high when the entire 64 bit of the data has been received by the transmitter providing an acknowledgement to the PC. The UART transmitter is

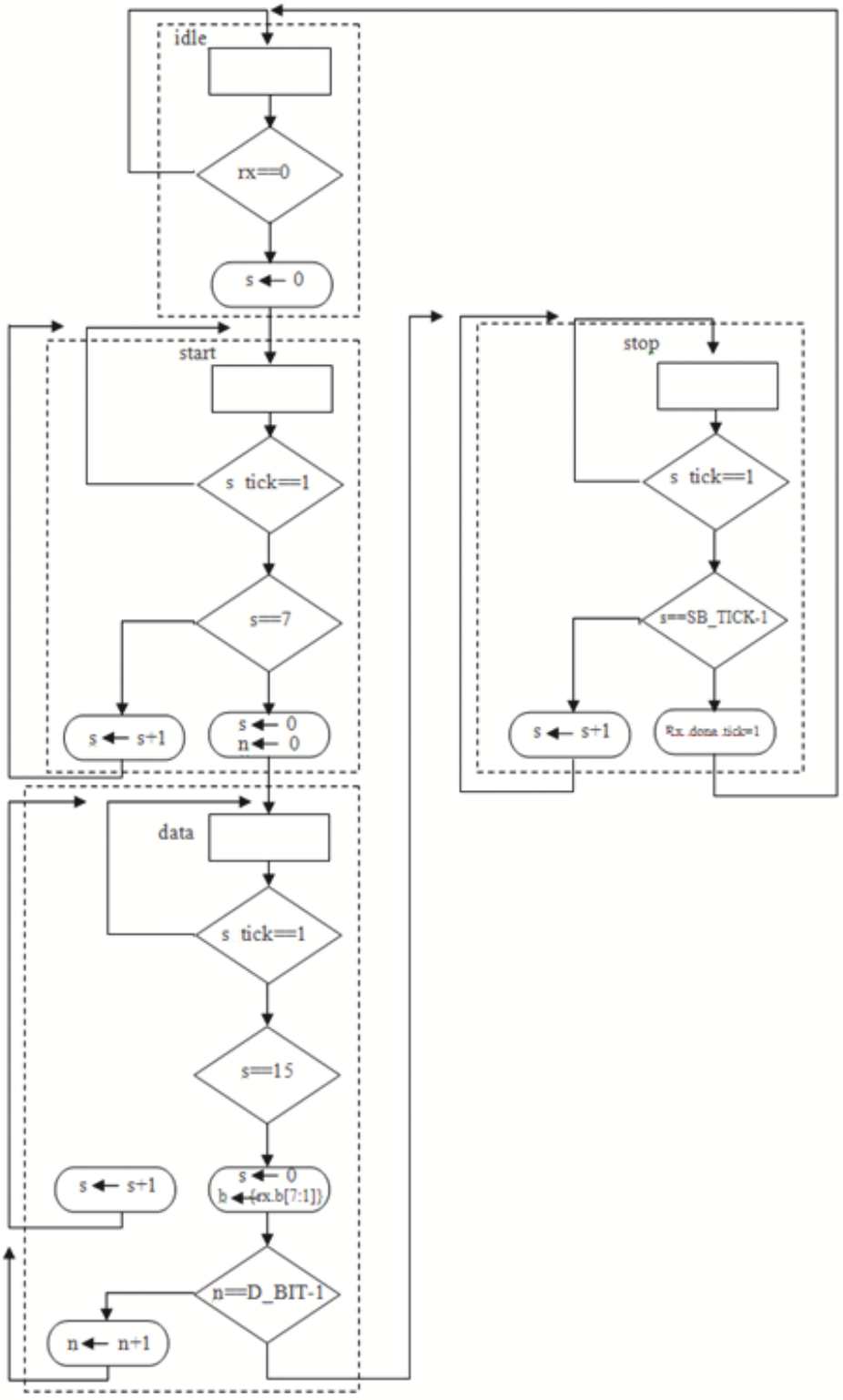

Figure 6.3: FSM state and functionality diagram of the receiver

also used to send results of arithmetic operations that the FPU performs to ensure proper operation; this was only used in the testing and verification stage.
Illustrated in figure 6.4 is the complete block diagram of the UART we have used in our design.

6.1.4 First In First Out (FIFO) buffer

The data to be transmitted by the PC is 64 bits long. 32 bits from it are composed of the IEEE754 standard of the actual data, where as 11 bits are used for the address of the memory where the data would be stored. 8 bits will be used for the master control switch while the remaining bits are free. See figure 6.5

The FIFO buffer is also used when we are transmitting instructions from the PC to be stored in the instruction memory that will later on access the data memory and perform the desired operation. The instruction memory contains 11 bits for the operand A, 11 bits for operand B, 11 bits for address of the destination for where to store the result, 3 bits of the operation to be performed, 13 bits of where to store the instruction, 5 bits free and 8 bits of control as shown in the figure 6.6

6.1.5 Hyper-Terminal

The hardware being configured and ready, the next step is to use a Windows based application or any other operating system of choice to access the com port. Com ports are communication ports assigned by the operating system to the serial port

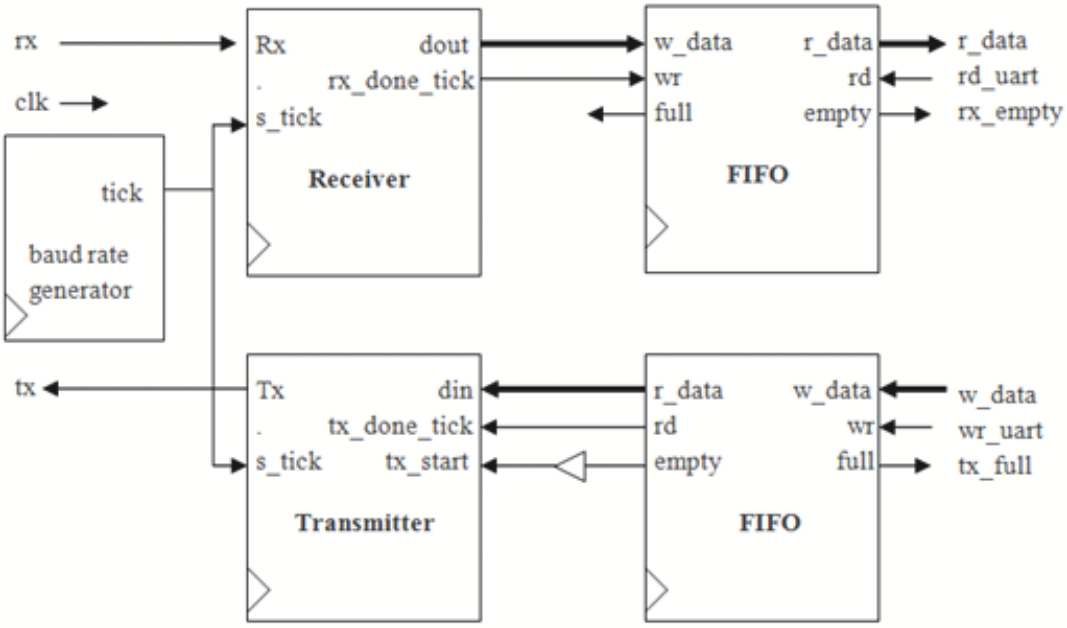

Figure 6.4: Complete UART system

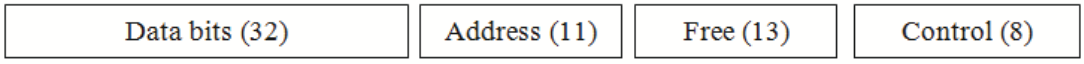

Figure 6.5: Structure of the Data RAM elements

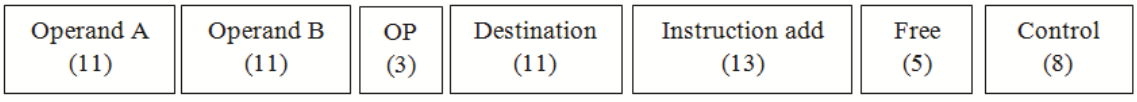

Figure 6.6: Structure of the Instruction RAM elements

or any other port that connects external hardware to the main PC. In Windows the HyperTerminal program is used to send and receive data through the serial port. In the more recent versions of Windows this program is not readily present but third party software like Tera Term can be used instead. The HyperTerminal or Tera Term software should be first configured in order for it to match the configurations of the UART. The configurations were set as 8 bits of data, 19200 baud rate, No bits of parity and 1 stop bit.

6.2 Selecting platform for software development

There are several options when developing a Windows based application with each having its pros and cons. Following is a brief reasoning for the choice of platform in application development. This analysis gives an insight to scenarios where certain platform can be advantageous in certain applications.

6.2.1 Java Development Kit (JDK)

The Java platform is a very powerful development tool that allows the developer to integrate many aspects of application development in one program. The measure of flexibility is great and the possible exports to mobile platforms and software defined hardware kits make it an attractive option. Java can be used to develop a wide range of applications that are mostly in the gaming and corporate application domain.

The disadvantage of Java is that certain scientific API's are not freely available

and in specific case of Digital Signal processing, the API's are intellectual properties of certain corporations [105]. Nevertheless when the application is only to interface a serial connected device, java development doesn't pose any problems. However, the reason Java was not used as the development environment is because it is very challenging to convert Java code to other platforms. There also exists a certain degree of learning curve in Java. JDK has its own programming environment and inheritances of APIs that many electrical engineers have never used. This limits the option for future development of the program if any enhancement is necessary.

6.2.2 C++

C++ is mainly used in device driver programming together with C language due to its fast compilation and strong hardware interfacing related advantages. The use of data structures and pointers give C++ an edge over Java in developing complex interfacing applications. Unfortunately this also introduces unnecessary complexities in the development of applications. C++ is also very difficult to migrate to other platforms and not very user friendly in GUI development [106]. Therefore it's a powerful but not an easy solution in context to our application requirement.

6.2.3 Matlab

Matlab has its own version of GUI development called GUIDE. This interactive and user friendly GUI development program is primarily used to interface externally connected devices. In the context of application for scientific purpose Matlab GUIDE proposes a few advantages that other platforms don't offer.

The primary advantage is the embedding of previously designed algorithms that were simulated tested on Matlab. The generated data sets using Matlab can directly be integrated with GUIDE to be sent serially to the FGPA board. The processed data can then be directly received from the FGPA and further computations can be performed on the data. This gives us great advantage as our device can be used as a standalone processor of modulation classification but if the need arises we can link the FPGA device with a computer and enhance the functionality accordingly.

6.3 Developed Graphical User Interface (GUI)

The developed graphical user interface is illustrated in figure 6.7. The GUI aids in connecting the personal computer (PC) to the FPGA board containing the modulation classification algorithm.

The user selects the baud rate and clicks the connect button. The GUI software then serially connects to the UART on the FPGA board. After the serial connection has been established, the user sends desired waveform data to the FPGA for training and testing. The training data remains the same for all 6 modula-

tion types, thus there is no option to explicitly transfer the training data. This is done automatically when the user wants to test the system. To test the system, the user sends a desired modulation scheme from the 'Modulation' list in the 'Signal' box. The SNR and fading effects are also selected optionally from the 'SNR' and 'Rayleigh Fading channel effects' lists respectively. After selecting the desired composition, the user clicks the 'Transmit Data button' to send the data to FPGA. The axis box shows the signal plots of the desired modulation schemes on the right side of the GUI.

After receiving all the data from the PC, the algorithm is run and tested on the FPGA and details of the processing are sent back to the GUI on the PC. The classified modulation scheme is turned green on the bottom right side of the GUI. Other parameters like the 'Total clock cycles', 'Total Time taken' and the 'Total Power consumed' are shown in the 'Results' box on the bottom left side of the GUI.

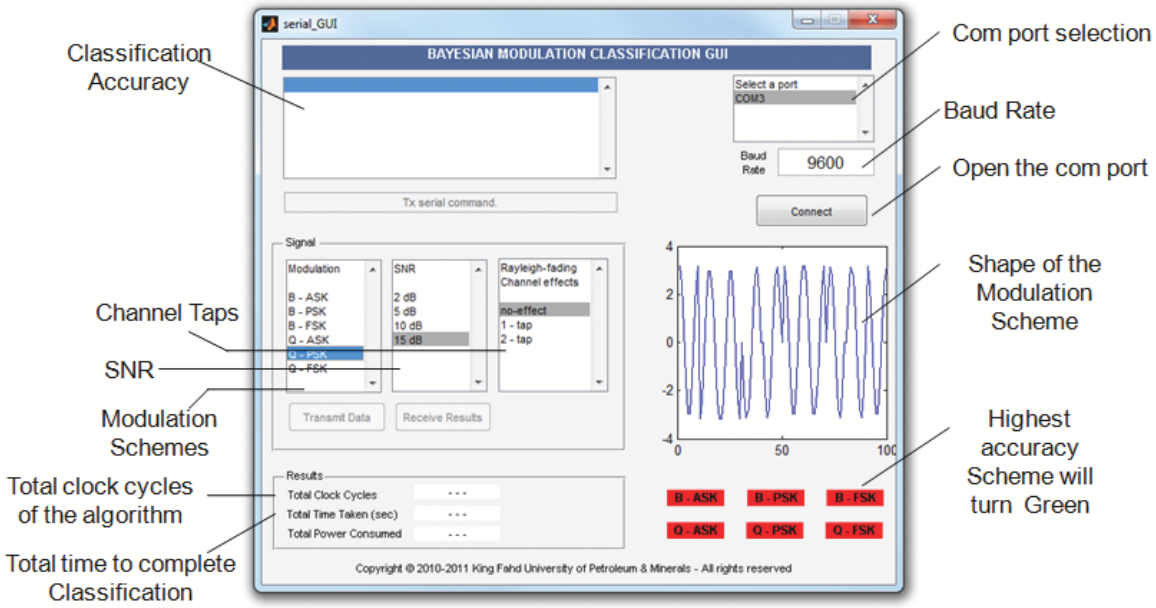

Figure 6.7: Designed GUI for the PC to FPGA interface

CHAPTER 7

CONCLUSIONS

The primary direction of the thesis was towards the classification of signals with unknown modulations and no a-priori knowledge using simple bayes classifier. Latter part of the thesis contained the hardware implementation of the proposed classification algorithm. The purpose of the implementation was to test the algorithm and evaluate it for real time performance.

A comprehensive literature study was performed of the general approaches used in the process of modulation classification. Different features and parameters used in the modulation classifiers were also discussed. The approaches proposed in the literature were reviewed under their respective categories. Their practical considerations and previously implemented systems and prototypes were also studied.

It was analyzed that not many of the modulation classifiers have been practically proven to work with signals of low SNR (below 8dB). It is uncommon to find a modulation classifier that performs well on low SNR and can also be realized into a compact hardware module. Thus, a tradeoff was observed between the classi-

fication accuracy and the architecture complexity of the hardware implemented modulation classifier. The proposed modulation classifier was aimed to accommodate for this necessity.

The thesis objectives previously defined were all met and completed. The proposed automatic modulation classifier and its hardware implementation were tested and evaluated successfully for real time performance. A short summary of the major contributions by this thesis are presented in the following section.

7.1 Summary of Contribution

Following is the list of contributions that were proposed in this thesis. All the following achievements were tested and verified.

- The proposed modulation classification algorithm performs with high accuracy in low SNR conditions
- The classification algorithm requires very small quantities of sampled data
- All operations of classification are performed in real time
- The proposed classifier identifies 6 modulation schemes
- The proposed classifier adapts to channel effects like path loss and multipath
- The proposed classifier does not depend on full a-priori knowledge of the signal

- A practical implementation of the modulation classifier was proposed on FPGA
- The architecture is generic and can be used for a large range of DSP algorithms
- The architecture holds low complexity even with large designs
- The architecture can performs multiple operations at the same time
- The architecture can be scaled up or down, very easily, w.r.t the application
- The hardware implementation contains a pipelined architecture that can be scheduled serially

7.2 Future Direction

Some future work can be directed towards this framework to make it more effective, as other modulation schemes can be introduced. It is hoped that more work can be carried out on the proposed framework in the following areas

1. **Multi Carrier Classification** While introducing other modulation schemes in the framework, one modulation scheme stands out from the rest. That is Orthogonal Frequency Division Multiplexing (OFDM). The classification of OFDM from other modulation schemes is quite simple due to the fact that there are multiple carriers in OFDM. These multiple carriers can be separated from the single carrier modulation schemes by just applying

a simple Gaussianity test. The test would be performed on the amplitude distribution of the sampled signal data.

2. **Faster performing FPU** In the proposed hardware implementation of the modulation classifier, the IEEE-754 floating point unit (FPU) is obtained from open source-ware websites. The obtained FPU lacks speed and area compactness. More research can be done to develop an FPU that performs faster and takes less space.

3. **Adaptive Scalability** One of the key features of the proposed hardware implementation is its scalability. The entire architecture can be scaled up or down according to the application or the resources. More research can be done to make this feature automatic or adaptive. The architecture can scale up or down automatically improving the efficiency or the speed by evaluating the resources.

7.3 Conclusion

The goal of this thesis is to design a practical modulation classification system to address the increasing needs of the software radio designers. Recognition of digital signal formats is an important application for software radio devices. Most of the proposed modulation classifiers are based on simulations and only cater for impairments of additive white Gaussian noise. However, in real world, most of the communication applications require the module to be compact and realizable.

Also, most of the communication channels are multi-path channels. As a result the communication signals face the effect dispersion.

The proposed framework presents an automatic digital modulation classification algorithm that classifies six modulation schemes in low SNR and Rayleigh fading channel environment. Firstly, a feature based classification algorithm is developed with four moments to identify within the schemes. The four features are chosen to be the first four moments. The Bayes classifier is selected to be the supervised learning algorithm in the proposed framework. A thresholding method is used in the post classification stage to further improve the accuracy. A zero crossing measure is used to identify between FSK-2 and FSK-4. Results show that the proposed classifier performs well at low SNR conditions. The classification accuracy at 2dB SNR is 93.8% for the 6 modulation schemes. Under the degrading conditions of Rayleigh fading channel, the classification accuracy at 10dB SNR becomes 75.3%. In comparison with the literature, this performance is very good. The proposed modulation classifier was also implemented on hardware for verification and real-time performance evaluation. The classification algorithm was developed with Matlab 2010b and implemented using on Xilinx Spartan3E-XC1200efg320 FPGA device. The implemented hardware modules were verified and validated for different modulation schemes. The results show that the hardware implementation closely matches the simulated behavior of the algorithm. The developed architecture is a general purpose core that can be used to test and verify a wide variety of DSP algorithms. The architecture holds low complexity

even with large designs by parallel processing and pipelined operations. The architecture can be scaled up or down, very easily, according the application or the resources at hand.

REFERENCES

[1] Shannon, C.E., "Communication in the Presence of Noise," *Proceedings of the IRE* , vol.37, no.1, pp. 10- 21, Jan. 1949

[2] R. F. Harrington, "Effect of antenna size on gain, bandwidth, and efficiency," *J. Res. Nat. Bureau Standards*, vol. 64D, pp. 1-12, Jan.-Feb. 1960

[3] Gardner, W., "Spectral Correlation of Modulated Signals: Part I–Analog Modulation," *IEEE Transactions on Communications*, vol.35, no.6, pp. 584- 594, Jun 1987

[4] S. G. Wilson, "Digital Modulation and Coding. Englewood Cliffs," *NJ: Prentice-Hall*, 1996.

[5] Murota, K.; Hirade, K., "GMSK Modulation for Digital Mobile Radio Telephony," *IEEE Transactions on Communications*, vol.29, no.7, pp. 1044- 1050, Jul 1981

[6] Mitola, J., III; Maguire, G.Q., Jr., "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol.6, no.4, pp.13-18, Aug 1999

[7] T. Gallaghan, J. Pery, and J. Tjho, "Sampling and algorithms aid modulation recognition," *Microwaves RF*, vol. 24, no. 9, pp. 117-119,, 1985.

[8] F. Jondral, "Automatic classification of high frequency signals," *Signal Processing*, vol. 9, no. 3, pp. 177-190, 1985.

[9] J. Aisbett, "Automatic modulation recognition using time domain parameters," *Signal Processing*, vol. 13, no. 3, pp. 323-328, 1987.

[10] L. Dominguez, J. Borrallo, J. Garcia, and B. Mezcua, "A general approach to the automatic classification of radio communication signals," *Signal Processing*, vol. 22, no. 3, pp. 239-250, 1991.

[11] E. Azzouz and A. Nandi, "Automatic modulation recognition-Part I," *J. Franklin Inst.*, vol. 334B, no. 2, pp. 242-273, 1997.

[12] A. Nandi and E. Azzouz,; "Modulation recognition using artificial neural networks," *Signal Processing*, vol. 56, pp. 165-175, 1997.

[13] Theodoridis S, Koutroumbas K, "Pattern recognition," *Academic Press*, London, 2009.

[14] W. Wei and J.M. Mendel, "A New Maximum-Likelihood Method for Modulation Classification," *Proceedings of ASILOMAR-29*, vol. 2, pp. 1132-1136, November 1995.

[15] D. Boiteau and C. Le Martret, "A General Maximum Likelihood Framework for Modulation Classification," *Proceedings of ICASSP*, vol. 4, pp. 2165-2168, May 1998.

[16] Y. Yang and C.H. Liu, "An asymptotic optimal algorithm for modulation classification," *IEEE Communication Letters*, vol. 2, no. 5, pp. 117-119, May 1998.

[17] Y. Yang and S.S. Soliman, "Optimum Classifier for M-ary PSK Signals," *IEEE International Conference on Communications*, vol. 3, pp. 1693-1697, June 1991.

[18] Y. Yang and S.S. Soliman, "A Suboptimal Algorithm for Modulation Classification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 38-45, January 1997.

[19] K. Kim and A. Polydoros, "Digital Modulation Classsification: The BPSK versus QPSK case," *Proceedings of IEEE MILCOM*, vol. 2, pp. 841-845, November 1991.

[20] A. Polydoros and K. Kim, "On the Detection and Classification of Quadrature Digital Modulations in Broad-Band Noise," *IEEE Transactions on Communication*, vol. 38, no. 8, pp. 1199-1211, August 1990.

[21] C.Y. Hwang and A. Polydoros, "Advanced Methods for Digital Quadrature and Offset Modulation Classification," *Proceedings of IEEE MILCOM*, vol. 2, pp. 841-845, November 1991.

[22] C.S. Long, K.M. Chugg and A. Polydoros, "Further Results in Likelihood Classification of QAM Signals," *Proceedings of IEEE MILCOM*, vol. 1, pp. 57-61, October 1994.

[23] Sills, J.A., "Maximum-likelihood modulation classification for PSK/QAM," *Proceedings of IEEE MILCOM*, pp. 57-61, 1999.

[24] Hong, L., and Ho, K.C., "Classification of BPSK and QPSK signals with unknown signal level using the Bayes technique," *Proceedings of IEEE ISCAS*, IV.1-IV.4, 2003.

[25] Panagiotou, P., Anastasoupoulos, A., and Polydoros, A., "Likelihood ratio tests for modulation classification," *Proceedings of IEEE MILCOM*, pp. 670-674, 2000.

[26] Lay, N., and Polydoros, A., "Per-survivor processing for channel acquisition, data detection and modulation classification," *Proceedings of ASILOMAR*, pp. 170-174, 1995.

[27] Lay, N., and Polydoros, A., "Modulation classification of signals in unknown ISI environments," *Proceedings of IEEE MILCOM*, pp. 170-174, 1995.

[28] De Vito, L., Rapuano, S., Villanacci, M., "An Improved Method for the Automatic Digital Modulation Classification," *Instrumentation and Measurement Technology Conference Proceedings*, IMTC, IEEE, pp. 1441-1446, May 2008.

[29] Wei Su; Xu, J.L.; Mengchu Zhou;, "Real-time Modulation Classification Based On Maximum Likelihood," *IEEE Communications Letters*, vol.12, no.11, pp.801-803, November 2008

[30] Jain, A.K., Duin, R.R.W., and Mao, J., "Statistical Pattern Recognition: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4-37, 2000.

[31] Azzouz, E.E., and Nandi, A.K., "Automatic modulation recognition of communication signals," *Kluwer Academic*, 1996.

[32] Hsue, S.Z., and Soliman, S.S., "Automatic modulation recognition of digitally modulated signals," *Proceedings of IEEE MILCOM*, pp. 645-649, 1989.

[33] Hsue, S.Z., and Soliman, S.S., "Automatic modulation classification using zero crossing," *IEEE Proceedings of Radar Signal Processing*, vol. 137, pp. 459-464, 1990.

[34] Ho, K.C., Prokopiw, W., and Chan, Y.T., "Modulation identification by the wavelet transform," *Proceedings of IEEE MILCOM*, pp. 886-890, 1995.

[35] Ho, K.C., Prokopiw, W., and Chan, Y.T., "Modulation identification of digital signals by the wavelet transform," *IEEE Proceedings of Radar & Sonar Navigation*, vol 47, pp. 169-176, 2000.

[36] Yang, Y., and Soliman, S.S., "Statistical Moments Based Classifier for MPSK Signals," *Proceedings of GLOBECOM*, pp. 72-76, 1991.

[37] Soliman, S.S., and Hsue, S.Z.: "Signal Classification using Statistical Moments," *IEEE Transactions on Communications*, vol. 40, pp. 908-916, 1992.

[38] Yang, Y., and Soliman, S.S., "An Improved Moment-Based Algorithm for Signal Classification," *Signal Processing*, vol. 43, pp. 231-244, 1995.

[39] Swami, A., and Sadler, B.M., "Hierarchical digital modulation classification using cumulants," *IEEE Transactions on Communication*, 48, pp. 416-429, 2000.

[40] Dai, W., Wang, Y., and Wang, J., "Joint power and modulation Classification using second- and higher statistics," Proc. WCNC, 2002, pp. 155-158

[41] Swami, A., Barbarossa, S., and Sadler, B., "Blind Source Separation and Signal Classification," Proceedings of ASILOMAR, pp. 1187-1191, 2000.

[42] Martret, C., and Boiteau, D.M., "Modulation classification by means of different order statistical moments," Proc. IEEE MILCOM, pp. 1387-1391, 1997.

[43] Marchand, P., Lacoume, J.L., and Le Martret, C., "Classification of linear modulations by a combination of different orders cyclic cumulants," Proc. Workshop on HOS, pp. 47-51, 1997.

[44] Marchand, P., Lacoume, J.L., and Le Martret, C., "Multiple hypothesis classification based on cyclic cumulants of different orders," Proc. ICASSP, pp. 2157-2160, 1998.

[45] Spooner, C.M., "Classification of cochannel communication signals using cyclic cumulants," Proc. ASILOMAR, pp. 531-536, 1995.

[46] Spooner, C.M., Brown, W.A., and Yeung, G.K., "Automatic radio-frequency environment analysis," Proc. ASILOMAR, pp. 1181-1186, 2000.

[47] Spooner, C.M., "On the utility of sixth-order cyclic cumulants for RF signal classification," Proc. ASILOMAR, pp. 890-897, 2001.

[48] Dobre, O.A., Bar-Ness, Y., and Su, W., "Higher-order cyclic cumulants for high order modulation classification," Proc. IEEE MILCOM, pp. 112-117, 2003.

[49] Dobre, O.A., Bar-Ness, Y., and Su, W., "Robust QAM modulation classification algorithm based on cyclic cumulants," Proc. WCNC, pp. 745-748, 2004.

[50] Dobre, O.A., Abdi, A., Bar-Ness, Y., Su, W, "Selection combining for modulation recognition in fading channels," *Proceedings of IEEE MILCOM*, pp. 2499-2505, 2005.

[51] Stolman, V.J., Paranjpe, S., and Orsak, G.C., "A blind information theoretic approach to automatic signal classification," *Proceedings of IEEE MILCOM*, pp. 447-451, 1999.

[52] Paris, B.-P., Orsak, G.C., Chen, H., and Warke, N., "Modulation classification in unknown dispersive environments," *Proceedings of ICASSP*, pp. 3853-3856, 1997.

[53] Mobasseri, B.G., "Constellation shape as a robust signature for digital modulation recognition," *Proceedings of IEEE MILCOM*, pp. 442-446, 1999.

[54] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "A survey of automatic modulation classification techniques: classical approaches and new trends," *Proc. IET Communications*, vol. 1, pp. 137-156, 2007.

[55] Huo, X., and Donoho, D.L., "A simple and robust modulation classification method via counting," *Proc. ICASSP*, pp. 3289-3292, 1998.

[56] Donoho, D.L., and Huo, X., "Large-sample modulation classification using Hellinger representation," *Proc. Signal Processing Advances in Wireless Communications*, pp. 133-137, 1997.

[57] Swami, A., and Sadler, B., "Modulation classification via hierarchical agglomerative cluster analysis," *Proc. IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications*, pp. 141-144, 1997.

[58] Schreyogg, C., "Identification of voiceband data signal constellations using a divisive cluster algorithm," *Proc. IEEE Digital Signal Processing Workshop*, pp. 474-477, 1996.

[59] Assaleh, K., Farrell, K.R., and Mammone, R.J., "A new method of modulation classification for digitally modulated signals," *Proc. IEEE MILCOM*, pp. 712-716, 1992.

[60] Farrell, K.R., and Mammone, R.J., "Modulation classification using a neural tree network," *Proc. IEEE MILCOM*, pp. 1028-1032, 1993.

[61] Liedtke, F.F., "Computer Simulation of an Automatic Classification Procedure for Digitally Modulated Communication Signals with Unknown Parameters," *Signal Processing*, vol. 6, pp. 311-323, 1984.

[62] Sapiano, P.C., Martin, J., and Holbeche, R., "Classification of PSK signals using the DFT of phase histogram," *Proc. ICASSP*, pp. 1868-1871, 1995.

[63] Schreyogg, C., Kittel, C., Kressel, U., and Reichert, J., "Robust classification of modulation types using spectral features applied to HMM," *Proc. IEEE MILCOM*, pp. 1377-1381, 1997.

[64] He Bing, Liu Gang, Ge Cun, Gao Jiang, "Modulation recognition of communication signal based on wavelet RBF neural network," *2nd International Conference on Computer Engineering and Technology (ICCET)*, vol. 2, pp. 490-492, April 2010.

[65] Joanes, D. N., Gill, C. A., "Comparing measures of sample skewness and kurtosis," *Journal of the Royal Statistical Society*, vol. 47, pp. 183-189, 1998.

[66] Hatzichristos, G., and Fargues, M.P., "A Hierarchical Approach to the Classification of Digital Modulation types in Multipath Environments," *Proceedings of ASILOMAR*, pp. 1494-1498, 2001.

[67] Ebrahimzadeh, A., "Automatic modulation recognition using RBFNN and efficient features in fading channels," *First International Conference on Networked Digital Technologies*, pp. 485-488, July 2009.

[68] Kim, K., Akbar, I.A., Bae, K.K., Jung-sun U., Spooner, C.M., Reed, J.H., "Cyclostationary Approaches to Signal Detection and Classification in Cognitive Radio," *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 212-215, April 2007.

[69] Grimaldi, D., Rapuano, S., De Vito, L., "An Automatic Digital Modulation Classifier for Measurement on Telecommunication Networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 56, pp. 1711-1720, Oct. 2007.

[70] Venalainen, J., Terho, L., and Koivunen, V., "Modulation classification in fading multipath channel," *Proc. ASILOMAR*, pp. 1890-1894, 2002.

[71] Louis, C., and Sehier, P., "Automatic Modulation Recognition with a Hierarchical Neural Network," *Proceedings of IEEE MILCOM*, pp. 713-717, 1994.

[72] Chung, C.D., and Polydoros, A., "Envelope-based Classification Schemes for Continous-Phase Binary FSK Modulations," *Proceedings of IEEE MILCOM*, pp. 796-800, 1994.

[73] Namjin Kim; Kehtarnavaz, N.; Yeary, M.B.; Thornton, S.;, "DSP-based hierarchical neural network modulation signal classification," *IEEE Transactions on Neural Networks*, vol.14, no.5, pp. 1065- 1071, Sept. 2003

[74] Cheol-Sun Park; Dae Young Kim; , "A novel robust feature of modulation classification for reconfigurable software radio," *IEEE Transactions on Consumer Electronics*, vol.52, no.4, pp.1193-1200, Nov. 2006

[75] Ren Guanghui; Zhao Yaqin; Wu Zhilu; Gu Xuemai;, "Channelized receiver platform of SDR based on FPGAs," *5th International Conference on Proceedings on ASIC*, vol.2, no., pp. 840- 843 Vol.2, 21-24 Oct. 2003

[76] Matic, V.; Pavic, B.; Tadic, V.; , "The implementation of digital signal processing for automatic recognition of radio emission type and spectrum occupancy analysis," *Devices, Circuits and Systems, Proceedings of the 2000 Third IEEE International Caracas Conference on*, pp.T55/1-T55/6, 2000

[77] Enrico Buracchini, "The Software Radio Concept," *IEEE Communication Magazine*, vol. 38, September 2000, pp. 138-143.

[78] John glossner, Daniel lancu, jinlu, erdens hokenede andMayan moudgill., "A software-defined Communications baseband design," *IEEE Communication Magazine*, vol. 41, pp.120-128, January 2003.

[79] Matic, V.; Lestar, B.; Tadic, V.; , "The use of digital signal processing for a modulation classification," *Electrotechnical Conference, 2002. MELECON 2002. 11th Mediterranean* , pp. 126- 130, 2002.

[80] P. Prakasam; M. Madheswaran;, " Reconfigurable Automatic Modulation Identification Hardware Module for Software Defined Radio Receivers," *International Journal of Computer Theory and Engineering*, Vol. 1, No. 5, December 2009.

[81] R. Srikanteswara, R. Chembil Palat, J. H. Reed and P. Athanas., "An Overview of Configurable Computing Machines for Software Radio Handsets," *IEEE communication Magazine*, vol. 41, pp.134-141, July 2003.

[82] R. Hartenstein, "A decade of reconfigurable computing: a visionary retrospective," *Proc. Int. Conf. and Exhibition on Design, Automation and Test in Europe*, pp. 642- 649, 2001.

[83] M. Cummings and S. Haruyama., "FPGA in the Software Radio," *IEEE Communication Magazine*, vol. 37, pp. 108- 112, Feb. 1999.

[84] Jun-Seo Lee, Jong-Hyun Park, Sang-Woo Kim,YingShan Li and Heung-Gyoon Ryu., "Implementation of DSP-Based Digital Receiver for the SDR Application," *Proc. l0th Asia-Pacific Conf. on Communications and 5th Int. Symposium on Mu1ti-Dimensional Mobile Communications*, pp 06-10, 2004.

[85] Huie. J, D'Antonio. P, Pelt. R and Jentz. B., "Synthesizing FPGA Cores for Software Defined Radio General Dynamics Decision Systems," *Software Defined Radio Technical Conference and Product Exposition*, November 2003.

[86] A. Iversen, J. Krstad, N. Taylor and K. Brown, "Classifcation of Communication Signals and Detection of Unknown Formats using Artifcial Neural Networks," *in Proceedings of the Symposium on Military Communications*, Italy, April 2005.

[87] Nolan, K., Doyle, L., O'Mahony, D. & Mackenzie, P. "Modulation Scheme Recognition Techniques for Software Radio on a General Purpose Processor

Platform," *in Proceedings the First Joint IEI/IEEE Symposium on Telecommunications System Research*, Dublin, 5pp, November 27th, 2001.

[88] Yaqin Zhao; Guanghui Ren; Zhi Zhong; , "Modulation Recognition of SDR Receivers Based on WNN," *Vehicular Technology Conference, VTC 2006-Spring. IEEE 63rd*, vol. 5, pp.2140-2143, 7-10 May 2006.

[89] N.Bjrsell, P.Daponte, L.De Vito, S.Rapuano, "Automatic signal recognition for a flexible spectrum managment," *Proc. of IEEE International Measurement Conference (IMEKO)*, Lisbon, Portugal, 6-11 Sept. 2009.

[90] Suris, J.; Recio, A.; Athanas, P.; , "Rapid radio: A framework for human-assisted signal classification and receiver implementation," *Aerospace & Electronics Conference (NAECON), Proceedings of the IEEE 2009 National*, pp.308-312, 21-23 July 2009.

[91] Recio, A.; Suris, J.A.; Athanas, P.; , "Automatic modulation classification for rapid radio deployment," *Rapid System Prototyping (RSP), 2010 21st IEEE International Symposium on* , vol., no., pp.1-7, 8-11 June 2010.

[92] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms using different performance metrics, " *Technical Report, Cornell University*, 2005.

[93] Clark, P. and Niblett, T., "The CN2 Induction Algorithm," *Machine Learning*, vol. 3, pp. 261-283, 1989.

[94] Langley, P., Iba, W. and Thompson, K., "An Analysis of Bayesian Classifiers," *Proceedings of the 10th National Conference on Artificial Intelligence*, pp. 223-228.

[95] I. Rish, "An empirical study of the nave bayes classifer," *Proceedings of IJCAI-01 workshop on Empirical Methods in AI*, pp. 41-46, Italy, 2001.

[96] Berkeley Design Technology, "Microprocessor Report," *http://www.bdti.com/MyBDTI/pubs/mic-out.pdf*, vol. 10, Dec. 1996.

[97] Jihong Liu; Deqin Liang;, "A Survey of FPGA-Based Hardware Implementation of ANNs," *Neural Networks and Brain, 2005. ICNN&B '05. International Conference on* , vol.2, no., pp.915-918, 13-15 Oct. 2005.

[98] Jidan AI-Eryani, "Floating Point Unit," *www.opencores.org/projects.cgi/web/fpu100/overview*, 2006.

[99] Christian Bontemps, Nour Meddahi, "Testing normality, A GMM Approach," *Journal of Econometrics*, vol. 124, pp. 149-186, January 2005.

[100] Jain, R., "Channel Models: A tutorial," *www.cs.wustl.edu/jain/cse574-08/ftp/ channel model tutorial.pdf.*

[101] Abdi, A.; Dobre, O.A.; Choudhry, R.; Bar-Ness, Y.; Wei Su; , "Modulation classification in fading channels using antenna arrays," *Military Communications Conference, 2004. MILCOM 2004. IEEE* , vol.1, no., pp. 211- 217 Vol. 1, 31 Oct.-3 Nov. 2004.

[102] El-Mahdy, A.E.; Namazi, N.M.; , "Classification of multiple M-ary frequency-shift keying signals over a Rayleigh fading channel," *Communications, IEEE Transactions on* , vol.50, no.6, pp.967-974, Jun 2002.

[103] Rudolf Usselmann,; "Open Floating Point Unit," *http://users.eecs.northwestern.edu/ ismail/courses/c92/fpu/fpu.pdf*, May 2000.

[104] Digilent inc.,; "Digilent Nexys2 Board Reference Manual," *www.digilentinc.com/Data/Products/NEXYS2/Nexys2_rm.pdf*, 2008.

[105] P. P. Chu, "FPGA Prototyping by Verilog Examples: Xilinx Spartan-3 Version," *Wiley Interscience*, 2008.

[106] Johnson, J., "GUI Bloopers: Don't and Do's for Software Developers and Web Designers," *Morgan Kaufman Publishers*, 2000.

[107] Scacchi, W.; , "Understanding the requirements for developing open source software systems," *Software, IEEE Proceedings*, vol.149, no.1, pp.24-39, Feb 2002

Vitae

- Muhammad Saad Sadiq
- Born in Karachi, Pakistan on September 5, 1986
- Received Bachelor of Science (B.S.) from Sir Syed University of Engineering and Technology, Karachi, Pakistan in 2008.
- Joined King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia as a Research Assistant in September 2008.
- Completed Master of Science (M.S.) in Electrical Engineering in June 2011.
- Email: *msaadsadiq@gmail.com*